



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| | | |
|--|----|---|
| (51) International Patent Classification ⁶ : G06F 3/00, 3/14 | A1 | (11) International Publication Number: WO 00/31613 (43) International Publication Date: 2 June 2000 (02.06.00) |
|--|----|---|

(21) International Application Number: PCT/IL99/00637

(22) International Filing Date: 25 November 1999 (25.11.99)

(30) Priority Data:
127293 26 November 1998 (26.11.98) IL

(71) Applicant (for all designated States except US): CREATOR LTD. [IL/IL]; Gush Etzion Street 13, 54030 Givat Shmuel (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): GABAI, Oz [IL/IL]; Klee Street 14, 62336 Tel Aviv (IL). GABAI, Jacob [IL/IL]; Klee Street 12, 62336 Tel Aviv (IL). SANDLERMAN, Nimrod [IL/IL]; Horgin Street 44, 52493 Ramat Gan (IL).

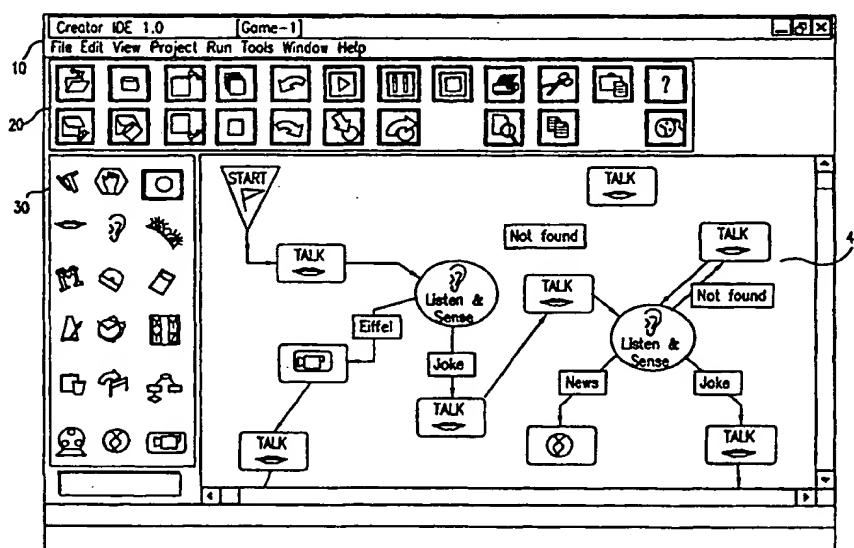
(74) Agents: COLB, Sanford, T. et al.; Sanford T. Colb & Co., P.O. Box 2273, 76122 Rehovot (IL).

(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), DM, EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

(54) Title: SCRIPT DEVELOPMENT SYSTEMS AND METHODS USEFUL THEREFOR



(57) Abstract

A system for generating scripts having verbal content (50), including a computer having a user input receiver operative to receive a user's definition of a script for at least one computer-controllable animated figure (120), the script including a plurality of interconnected script elements (40) each representing an action performable by the computer-controllable animated figure (130). A graphics interface operative to generate a pictorial image (140) of the script as the script is generated by the user, the graphics interface including a drag and drop facility operative to drag and drop script elements (30) and a flowchart generating facility.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|----|--------------------------|----|--|----|--|----|--------------------------|
| AL | Albania | ES | Spain | LS | Lesotho | SI | Slovenia |
| AM | Armenia | FI | Finland | LT | Lithuania | SK | Slovakia |
| AT | Austria | FR | France | LU | Luxembourg | SN | Senegal |
| AU | Australia | GA | Gabon | LV | Latvia | SZ | Swaziland |
| AZ | Azerbaijan | GB | United Kingdom | MC | Monaco | TD | Chad |
| BA | Bosnia and Herzegovina | GE | Georgia | MD | Republic of Moldova | TG | Togo |
| BB | Barbados | GH | Ghana | MG | Madagascar | TJ | Tajikistan |
| BE | Belgium | GN | Guinea | MK | The former Yugoslav Republic of Macedonia | TM | Turkmenistan |
| BF | Burkina Faso | GR | Greece | ML | Mali | TR | Turkey |
| BG | Bulgaria | HU | Hungary | MN | Mongolia | TT | Trinidad and Tobago |
| BJ | Benin | IE | Ireland | MR | Mauritania | UA | Ukraine |
| BR | Brazil | IL | Israel | MW | Malawi | UG | Uganda |
| BY | Belarus | IS | Iceland | MX | Mexico | US | United States of America |
| CA | Canada | IT | Italy | NE | Niger | UZ | Uzbekistan |
| CF | Central African Republic | JP | Japan | NL | Netherlands | VN | Viet Nam |
| CG | Congo | KE | Kenya | NO | Norway | YU | Yugoslavia |
| CH | Switzerland | KG | Kyrgyzstan | NZ | New Zealand | ZW | Zimbabwe |
| CI | Côte d'Ivoire | KP | Democratic People's Republic of Korea | PL | Poland | | |
| CM | Cameroon | KR | Republic of Korea | PT | Portugal | | |
| CN | China | KZ | Kazakstan | RO | Romania | | |
| CU | Cuba | LC | Saint Lucia | RU | Russian Federation | | |
| CZ | Czech Republic | LI | Liechtenstein | SD | Sudan | | |
| DE | Germany | LK | Sri Lanka | SE | Sweden | | |
| DK | Denmark | LR | Liberia | SG | Singapore | | |
| EE | Estonia | | | | | | |

SCRIPT DEVELOPMENT SYSTEMS AND METHODS USEFUL THEREFOR

FIELD OF THE INVENTION

The present invention relates to apparatus and methods for generating verbal content and for controlling toys and other manipulable objects.

BACKGROUND OF THE INVENTION

Technologies and systems related to the subject matter of the present application are described in the following documents:

Published PCT Application No. W097/18871 (PCT/IL96/00157); and in U.S. Patent Application Nos. 09/062,500 and 09/081,255, both to Gabai et. al., and U.S. Patent 5,752,880 to Gabai et. al.;

"Using the Microsoft Agent Character Editor", is available at following URL: http://premium.microsoft.com/msdn/library/sdkdoc/msagnet/deschar2zip_8r8x.htm;

"Designing characters for Microsoft Agent", is available at following URL: http://premium.microsoft.com/msdn/library/sdkdoc/msagnet/deschar2zip_67zl.htm;

Director, by Macromedia, is described at the following URL: <http://www.macromedia.com/software/director/productinfo/wha-tis.html>;

Creator, by LEGO Media International is described at the following URL: <http://www.legomedia.com/creator/creatorscreenshots2.asp>;

UniTools by Unitel Inc. is available at the following URL: <http://unitel.inc.com/website/unitools.htm>;

The disclosures of all publications mentioned in the specification and of the publications cited therein are hereby incorporated by reference.

SUMMARY OF THE INVENTION

The present invention seeks to provide apparatus and methods for generating scripts having verbal contents.

There is thus provided in accordance with a preferred embodiment of the present invention a system for generating scripts having verbal content, the system including a

computer having a user input receiver operative to receive a user's definition of a script for at least one computer-controllable animated figure, the script including a plurality of interconnected script elements each representing an action performable by the computer-controllable animated figure, the script including at least one verbal script element representing a verbal action performable by the computer-controllable animated figure, and a graphics interface operative to generate a pictorial image of the script as the script is generated by the user, the graphics interface including a drag and drop facility operative to drag and drop script elements and a flowchart generating facility, the facilities being operative to provide a flowchart of script elements including illustrated connections, having respective illustrated directions, interconnecting illustrations of each dragged and dropped script element, and a verbal output generator controlled by the user's definition of the script and operative to generate verbal output as defined in the script.

Further in accordance with a preferred embodiment of the present invention the animated figure includes a virtual animated figure on a computer screen.

Still further in accordance with a preferred embodiment of the present invention the animated figure includes a physical animated figure.

Additionally in accordance with a preferred embodiment of the present invention the physical animated figure is wirelessly connected to the computer.

There is further provided in accordance with another preferred embodiment of the present invention a script definition method including accepting a user's definition of a speech recognition script element including a designation of a plurality of keywords to be recognized, and providing a flowchart generation user interface operative to facilitate generation of a flowchart by a user representing a script defined by the user, the flowchart including a conditional flowchart element representing the speech recognition script element, the providing step including prompting the user to associate each branch of the conditional flowchart element with one of the plurality of keywords.

There is also provided in accordance with another preferred embodiment of the present invention a method for generating a computerized script including a speech recognition script element, the method including generating a speech recognition script element including selecting n keywords to be recognized, and generating a script junction having at least $n + 1$ branches corresponding respectively to the n keywords to be recognized and to a "not found" option which is followed if none of the n keywords are recognized in a user's utterance.

Further in accordance with a preferred embodiment of the present invention the method includes automatically running the script.

There is also provided in accordance with another preferred embodiment of the present invention a script generating method including providing a flowchart generation user interface operative to facilitate generation of a flowchart by a user representing a script defined by the user, the script including a plurality of script elements represented respectively by a plurality of flowchart elements, and using a computer to run the script defined by the user including highlighting, for the user, the flowchart element corresponding to the script element currently being executed.

Further in accordance with a preferred embodiment of the present invention the script is defined for an animated figure capable, in conjunction with the computer, of executing each of the script elements and wherein the using step includes activating the animated figure by running at least a portion of the script on the computer.

Still further in accordance with a preferred embodiment of the present invention the computer includes a simulator operative to simulate at least a portion of the script without utilizing the animated figure.

Additionally in accordance with a preferred embodiment of the present invention the computer includes a computer speaker and wherein the simulator is operative to simulate speech of the animated figure by activating the computer speaker.

Further in accordance with a preferred embodiment of the present invention the computer includes a computer microphone and wherein the simulator is operative to simulate reception of speech by the animated figure by activating the computer microphone.

Still further in accordance with a preferred embodiment of the present invention the computer includes a computer screen and a stored representation of the figure's performance of individual physical-action script elements and wherein the simulator is operative to simulate physical action of the animated figure by displaying, on the computer screen, an image of at least a portion of the figure performing the physical action.

Additionally in accordance with a preferred embodiment of the present invention the system also includes an Internet searcher.

Further in accordance with a preferred embodiment of the present invention the computer includes at least one input device and wherein the simulator is operative to simulate sensing activities of the animated figure by receiving mock sensory inputs via the input

device.

There is also provided in accordance with another preferred embodiment of the present invention a computerized script generating system including a user input receiver operative to receive a user's definition of a script including a plurality of interconnected script elements each representing an action performable by a computer-controllable device, and a script implementer receiving the user's definition of the script and operative to generate an integrated command, for transmission to the device, which command combines at least two adjacent script elements.

Further in accordance with a preferred embodiment of the present invention the script implementer is operative, for at least one pair of first and second script elements, to generate a single command, for transmission to the device, when the first and second script elements occur adjacently in the script and to generate a separate command for each of the first and second script elements when the first and second script elements occur separately rather than adjacently in the script.

There is also provided in accordance with another preferred embodiment of the present invention a system for generating scripts having verbal content, the system including a computer having a user input receiver operative to receive a user's definition of a script for at least one computer-controllable physical animated figure, the script including a plurality of interconnected script elements each representing an action performable by the computer-controllable animated figure, the script including at least one verbal script element representing a verbal action performable by the computer-controllable animated figure, and a verbal output generator controlled by the user's definition of the script and operative to generate verbal output as defined in the script, wherein the computer includes a simulator operative to simulate at least a portion of the script without utilizing the animated figure.

Further in accordance with a preferred embodiment of the present invention the system also includes a computerized scheduler operative to control the animated figure according to a user-determined schedule received by the scheduler.

Still further in accordance with a preferred embodiment of the present invention the system also includes an e-mail function.

Additionally in accordance with a preferred embodiment of the present invention the e-mail function includes apparatus for reading e-mail and apparatus for sending e-mail.

Further in accordance with a preferred embodiment of the present invention the system

is operative to generate a computer executable file operative to manipulate the toy without recourse to generation of scripts.

There is also provided in accordance with another preferred embodiment of the present invention a toy content development system for developing content for interactive computerized toys, the development system including for each of at least one computerized toy, a preprogrammed library of software functions which are executable by the toys, and a computing environment within which a user of the system can program at least one of the computerized toys by generating at least one combination of software functions in the library.

Further in accordance with a preferred embodiment of the present invention the combination includes a logical combination.

There is additionally provided in accordance with another preferred embodiment of the present invention a visual toy development system providing a visual environment for developing toy content, the system including a visual computing environment within which a user of the system can visually program at least one computerized toy by generating at least one combination of software functions, the environment being operative to provide a library of visual software functions which are executable by at least one toy.

Further in accordance with a preferred embodiment of the present invention the environment includes a preprogrammed library of visual software functions.

There is also provided in accordance with another preferred embodiment of the present invention a verbal interaction development system providing a visual environment for developing a verbal interaction, the system including a visual computing environment within which a user of the system can visually program a verbal interaction, the visual computing environment including a visual representation of at least one of a speech comprehension function and a speech generation function.

There is further provided in accordance with another preferred embodiment of the present invention a verbal interaction development system providing a visual environment for developing a verbal interaction, the system including a visual computing environment within which a user of the system can visually program a verbal interaction, the visual computing environment including a visual representation of at least one of a speech recognition function and a speech articulation function.

Further in accordance with a preferred embodiment of the present invention the visual computing environment is operative to accept a user's definition of at least one of the

following logical structures: a decision tree, a flowchart, and a state diagram, and to generate a manipulable visual representation thereof, wherein the visual representation includes a simulation of performance, by the toy, of the combination of software functions generated by the user, the simulation including a traversal of at least one of the logical structures.

There is also provided in accordance with another preferred embodiment of the present invention a verbal interaction development system providing a visual environment for developing a verbal interaction, the system including a computing environment within which a user of the system can program a verbal interaction, the computing environment including a representation of at least one of a speech recognition function and a speech articulation function, the speech recognition function including a speech template generator operative to accept, for at least one speech input, a user's definition of a set of templates, to each of which the speech input is to be compared, and a template recognition probability evaluator operative to evaluate the probability of correct matching of at least one speech input to each of the templates in the set.

Further in accordance with a preferred embodiment of the present invention the system also includes a developed content simulator operative to generate, on a screen display, a simulation of a programmed computerized toy performing a plurality of software functions in accordance with the programming of the toy by the user, using the computing environment.

Still further in accordance with a preferred embodiment of the present invention the system also includes a user workstation serving the user which is located remotely relative to the preprogrammed library and the computing environment.

Additionally in accordance with a preferred embodiment of the present invention the remotely located user workstation is connected to the preprogrammed library and the computing environment via the Internet.

Further in accordance with a preferred embodiment of the present invention the remotely located user workstation includes an Internet browser.

Still further in accordance with a preferred embodiment of the present invention the apparatus is substantially as shown and described above.

Additionally in accordance with a preferred embodiment of the present invention the apparatus is substantially as illustrated in any of the drawings.

Further in accordance with a preferred embodiment of the present invention the method is substantially as shown and described above.

Still further in accordance with a preferred embodiment of the present invention the method is substantially as illustrated in any of the drawings.

Preferably, the stored representation includes a plurality of graphic representations of the figure performing a corresponding plurality of script elements.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Fig. 1 is a screen display generated by an interactive script development system constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 2 is a screen display in which a script has been generated by dragging and dropping object types from the toolbox 30 into the work area 40;

Fig. 3 is a display screen showing characteristics of a TALK/PLAY script component, in which a toy participant called Johnny says "Hello my name is Johnny";

Fig. 4 is another screen display including a work area in which another script is represented;

Fig. 5 is a display screen showing characteristics of a TALK/PLAY script component, in which Johnny asks a question, namely: "Do you want to hear a joke or hear the news";

Fig. 6 is a display screen showing characteristics of a LISTEN AND SENSE script component, in which the system listens for one of the following keywords: "joke" or "picture";

Fig. 7 is a display screen showing characteristics of a TALK/PLAY script component, in which Johnny tells a joke;

Fig. 8 is a display screen showing characteristics of a DISPLAY GRAPHICS script component;

Fig. 9 is a display screen which is shown when a script creator selects TEXT;

Fig. 10 is a display screen showing fields via which properties of a selected END script component may be defined;

Fig. 11 is a display screen showing fields via which properties of a selected RECORD script component may be defined;

Fig. 12 is a display screen showing fields via which properties of a selected TALK

script component may be defined;

Fig. 13 is a display screen showing fields via which properties of a selected LISTEN AND SENSE script component may be defined;

Fig. 14 is a display screen showing fields via which properties of a selected MOVE script component may be defined;

Fig. 15 is a display screen showing fields via which properties of a selected LOCAL MEMORY script component may be defined;

Fig. 16 is a display screen showing fields via which properties of a selected GLOBAL MEMORY (STORAGE) script component may be defined;

Fig. 17 is a display screen showing fields via which properties of a selected CALCULATE script component may be defined;

Fig. 18 is a display screen showing fields via which properties of a selected WAIT script component may be defined;

Fig. 19 is a display screen showing fields via which properties of a selected TIME MARKER script component may be defined;

Fig. 20 is a display screen showing fields via which properties of a selected BOOLEAN (CONDITION) script component may be defined;

Fig. 21 is a display screen showing fields via which properties of a selected EXECUTE script component may be defined;

Fig. 22 is a display screen showing fields via which properties of a selected GOTO (JUMP) script component may be defined;

Fig. 23 is a display screen showing fields via which properties of a selected CALL SCRIPT script component may be defined;

Fig. 24 is a display screen showing fields via which properties of a selected RANDOM script component may be defined;

Fig. 25 is a display screen showing fields via which properties of a selected INTERNET script component may be defined;

Fig. 26 is a display screen showing fields via which properties of a selected GRAPHICS script component may be defined;

Fig. 27 is a display screen showing fields via which a script creator may define properties of a SCHEDULER option, selected by means of a Scheduler control button in control button array 20;

Fig. 28 is a display screen of a Project Manager option which may for example be selected by pushing a manager button located hierarchically below the Project command in the command bar 10;

Fig. 29 is a display screen which is typically generated during run time of a script, responsive to a command from a script creator who wishes to monitor the operation of the toys executing the script;

Figs. 30A - 30B, taken together, form a flowchart illustration of a main script for operating a toy called Storyteller, constructed and operative in accordance with a preferred embodiment of the present invention, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1;

Figs. 31A - 31B, taken together, form a flowchart illustration of the Game script called by the main script of Figs. 30A - 30B, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1;

Figs. 32A - 32B, taken together, form a flowchart illustration of the Song script called by the main script of Figs. 30A - 30B, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1;

Figs. 33A - 33B, taken together, form a flowchart illustration of the Story script called by the main script of Figs. 30A - 30B, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1;

Fig. 34 is a screen display of a main toy definition and testing menu;

Fig. 35 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Talk option in the main menu of Fig. 34;

Fig. 36 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Listen option in the main menu of Fig. 34;

Fig. 37 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Dialog option in the main menu of Fig. 34;

Fig. 38 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Move option in the main menu of Fig. 34;

Fig. 39 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Sense option in the main menu of Fig. 34;

Fig. 40 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Record option in the main menu of Fig. 34;

Fig. 41 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Hardware option in the main menu of Fig. 34;

Fig. 42 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the "more" option in the menu of Fig. 41;

Fig. 43 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Script option in the main menu of Fig. 34;

Fig. 44 is a screen display which is displayed to a toy defining user response to selection by the toy defining user of the Listen screen object in the screen display of Fig. 43;

Fig. 45 is a screen display which is displayed to a toy defining user response to selection by the toy defining user of the Talk screen object in the screen display of Fig. 43;

Fig. 46 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Setting option in the main menu of Fig. 34;

Fig. 47 is a screen display which is displayed to a toy defining user who seeks to combine a plurality of interconnected script elements each representing an action performable by the toy;

Fig. 48 is a semi-pictorial semi-block diagram illustration of a script development system constructed and operative in accordance with a preferred embodiment of the present invention in which the toy actuated by the script is a physical toy;

Fig. 49 is a semi-pictorial semi-block diagram illustration of a script development system constructed and operative in accordance with a preferred embodiment of the present invention in which the toy actuated by the script is a virtual toy represented on a screen display;

Fig. 50 is a pictorial illustration of a script developer running a script;

Fig. 51 is a screen display of a time scheduler forming part of the system, according to a preferred embodiment of the present invention;

Fig. 52 is a screen display of an Internet searcher constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 53 is a pictorial diagram of the Living Object base station;

Fig. 54 is a pictorial diagram of the Living Object Toy;

Fig. 55 is a screen display of the Scriptwriter icon on desktop;

Fig. 56 is a screen display of the Living Object Scriptwriter main Screen;

Fig. 57 is a screen display showing the "select tools – options" window;

- Fig. 58 is a screen display showing the "toy" window;
- Fig. 59 is a screen display showing the "Hardware" window;
- Fig. 60 is a screen display showing the "talk icon";
- Fig. 61 is a screen display of the Scriptwriter main window with the added talk object;
- Fig. 62 is a screen display of the Scriptwriter main screen display with the added talk object connected by a line to the start object;
- Fig. 63 is a screen display of the action toolbar with the save icon;
- Fig. 64 is a screen display for naming and saving the script;
- Fig. 65 is a screen window display of a combo box for typing the toy's speech;
- Fig. 66 is a screen window display for recording sound to be played by the toy;
- Fig. 67 is a screen window display for saving the recording;
- Fig. 68 is a screen window display for selecting a "wave" file to be played by the toy;
- Fig. 69 shows the Listen icon;
- Fig. 70 is a screen display of a part of the Scriptwriter main window with the Listen object added;
- Fig. 71 is the "Listen and Sense" screen window display;
- Fig. 72 is the "Keyword link box" in the "Choose Link" screen display;
- Fig. 73 is the Scriptwriter main screen display with a Listen object links to corresponding Talk objects;
- Fig. 74 is the Run-Run screen window display;
- Fig. 75 is the Sample error message screen window display;
- Figs. 76A and 76B, taken together, show a table of the functions provided by the Scriptwriter with their icons;
- Fig. 77 is the Talk screen window display;
- Fig. 78 is the Listen and Sense screen window display;
- Fig. 79 is the Move screen window display;
- Fig. 80 is the Record Options screen window display;
- Fig. 81 showing the Memory screen window display;
- Fig. 82 is the Condition screen window display;
- Fig. 83 is the Calculation screen window display;
- Fig. 84 is the Random screen window display;
- Fig. 85 is the Date and Time screen window display;

- Fig. 86 is the Wait screen window display;
- Fig. 87 is the Junp screen window display;
- Fig. 88 is the Execute screen window display;
- Fig. 89 is the Run Script screen window display;
- Fig. 90 is the Internet screen window display;
- Fig. 91 is the Graphics screen window display;
- Figs. 92, 93, 94, 95 and 96 are the corresponding "End", "Script Properties", "Choose Link", "Popup Menu" and the "Options" screen window displays;
- Fig. 97 is the Artificial Life Algorithm Editor screen window display;
- Fig. 98 is the Artificial Life Editor screen window display;
- Fig. 99 is the Artificial Life Editor screen window display with the Cell Management popup window;
- Fig. 100 is the Artificial Life Editor screen window display with the Function Library popup window;
- Fig. 101 showing the Artificial Life Manager screen window display;
- Fig. 102 is the Artificial Life Editor Viewer window display;
- Figs. 103 and 104 are the Scriptwriter main screen display with corresponding game script and laugh script;
- Fig. 105 is the Scriptwriter main screen display with the File menu open;
- Fig. 106 is the Scriptwriter main screen display with the Create Report Window;
- Fig. 107 is the Scriptwriter main screen display with the Edit menu open;
- Fig. 108 is the Scriptwriter main screen display with the Find Window;
- Fig. 109 is the Scriptwriter main screen display with the Replace Window;
- Fig. 110 is the Scriptwriter main screen display with the View menu open;
- Fig. 111 is the Scriptwriter main screen display with the Volume and Speech Recognition Windows;
- Fig. 112 is the Scriptwriter main screen display with the Watch List and the Add Memory windows;
- Fig. 113 is the Scriptwriter main screen display with the Execute Log and the Messages windows;
- Fig. 114 is the Scriptwriter main screen display with the Sensor Selection window;
- Fig. 115 showing the Scheduler screen window display;

Figs. 116 and 117 showing the Scheduler screen window display with the Add Task popup window Scheduler List popup window correspondingly;

Fig. 118 is the Scriptwriter main screen display with the Find Toys and Assign Channels window;

Fig. 119 is the Scriptwriter main screen display with the Run menu open;

Fig. 120 is the Scriptwriter main screen display with the Option window at the Toys Page;

Fig. 121 is the Scriptwriter main screen display with the Option window at the Hardware Page;

Fig. 122 is the Scriptwriter main screen display with the Option window at the Environment Page;

Fig. 123 is the Scriptwriter main screen display with the Option window at the Volume Setting Page;

Fig. 124 is the Scriptwriter main screen display with the Option window at the Living Toy Page; and

Fig. 125 is the Scriptwriter main screen display with the Option window at the Script page.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

The present specification uses the following terminology:

Speech recognition: Word-level interpretation of incoming speech data.

Speech comprehension: Phrase-level, sentence-level or idea-level interpretation of incoming speech data, preferably including emotional analysis.

Speech articulation: Word-level production of outgoing speech.

Speech generation: Phrase-level, sentence-level or idea-level production of outgoing speech data, preferably incorporating emotional content in the outgoing speech data.

Fig. 1 is a screen display generated by an interactive script development system constructed and operative in accordance with a preferred embodiment of the present invention.

As shown, the screen display of Fig. 1 includes a command bar 10, a control button array 20, a toolbox 30 and a work area 40. The command bar provides conventional file manipulation and project management options as well as editing tools. The control button array 20 preferably provides shortcuts to the commands in the command bar. The toolbox 30 preferably comprises a plurality of icons representing object types or script components for selection by a script creator (i.e. a human being using the system of Fig. 1 to create a script). Script component selection is typically effected by means of a conventional drag and drop operation, thereby to generate a script comprising a plurality of objects, typically associated with one another in flowchart form. A visual representation of the script is generated in the work area 40.

Fig. 1 is an initial screen display in which the work area 40 is empty. In Fig. 2, a script has been generated by dragging and dropping object types from the toolbox 30. As shown, the script comprises an initial verbalization 50, a subsequent listening process 60, and three subsequent verbalizations 70, 80 and 90, activated depending on the user's vocalization, as spotted in the course of listening process 60. In the illustrated embodiment, the contents of the verbalizations are not shown in the work area except responsive to a user request therefor e.g. by double-clicking on the icon representing an individual verbalization.

However, alternatively, verbalization contents, as well as other features of each verbalization, may be displayed without waiting for a user request. For example, the identity of the speaker may be represented, for each verbalization, by displaying an icon of the speaker and a bubble in which the verbalization contents appears. If the verbalization contents is determined by selection of a wavefile rather than by keying in text, the user may be prompted to provide a title or precis of the wavefile for display on the screen. For example, a wavefile of Martin Luther King's "I have a dream" speech may be selected by a user in which case a bubble containing the words "I have a dream..." may be displayed on the screen, preferably in

association with an icon of Martin Luther King.

In Fig. 3, for example, the script creator is prompted to assign a name to each talk object, and once this is done, the talk object may be displayed in association with its name.

The object types or script components provided in toolbox 30 may for example comprise the following:

START -- of script

END -- of script

RECORD -- record user, preserve recording for later rendering of recording

TALK/PLAY -- utter an input text or play a waveform file

LISTEN AND SENSE -- listen to human player, typically in order to detect one of several script-creator-determined speech options and/or in order to determine that none of the speech options selected by the script-creator have been identified. "Sense" refers to operation of any sensor other than a microphone, such as a microswitch, motion sensor, light sensor, temperature sensor, and so on.

MOVE/MANIPULATE -- body parts of toy participants in the script.

STORE IN LOCAL MEMORY -- information such as a numerical value, for subsequent use later in the script.

STORE IN GLOBAL MEMORY -- such as a hard disk, information such as a numerical value, for subsequent use later in the same execution of the script and/or in executions of other scripts and/or in a subsequent execution of the same script.

CALCULATE -- Perform a computation.

WAIT -- Wait.

TIME MARKER -- in order to compute elapsed time subsequently in the same script, or in other scripts.

BOOLEAN OPERATOR -- for generating conditional branching where the condition comprises a Boolean combination of simpler conditions.

EXECUTE A PROGRAM -- load and execute any computer program such as a wordprocessor program.

GOTO -- another script or another point within a current script.

CALL -- another script, thereby generating a hierarchical relationship between the calling and called scripts.

RANDOM -- selection of one option within a script-creator-defined set of options.

ACCESS INTERNET -- Access a network such as the Internet.

DISPLAY GRAPHICS -- Display an image or a video sequence, on the screen.

TEXT -- Comment which is displayed but is not part of the flowchart.

Fig. 4 is another screen display including a work area in which another script is represented. The script of Fig. 4 includes a first question (100) to a human player, to which the human player is anticipated to respond either with the word "Eiffel" or "joke", indicating respectively a desire to view an image of the Eiffel tower on the computer display and a desire to hear a joke. If the human player responds with "Eiffel" or "joke" respectively, the computer displays (120) the Eiffel image or plays (130) a joke respectively. The script of Fig. 4 also includes a second question (140) to the human player, to which the human player is anticipated to respond either with the word "news" or "joke", indicating respectively a desire to see a news broadcast on the computer display and a desire to hear a joke. If the human player responds with "news" or "joke" respectively, the computer plays (160) the news or plays a joke (170) respectively. The news may be radio, television or Internet news. News may be played either by generating an on-line connection to news being currently broadcast, or by replaying a most recently stored news edition.

Preferably, when the script creator drags and drops the "listen and sense" script component, the system automatically generates a "not found" exit from "listen and sense". For example, if the script creator generates two exits from "listen and sense", such as "Eiffel" and "joke", the system typically generates $2 + 1 = 3$ exits, the third exit being "not found". The script creator is thereby prompted to determine the eventuality, if none of the anticipated player inputs are sensed.

Fig. 3 is a display screen showing characteristics of a TALK/PLAY script component, in which a toy participant called Johnny says "Hello my name is Johnny". The script component of Fig. 3 includes the identity of the speaker, the source of the desired audio output (text rather than wavefile), and so on.

Fig. 5 is a display screen showing characteristics of a TALK/PLAY script component, in which Johnny asks a question, namely: "Do you want to hear a joke or hear the news". The script component of Fig. 5 includes the identity of the speaker, the source of the desired audio output (text rather than wavefile), and so on.

Fig. 6 is a display screen showing characteristics of a LISTEN AND SENSE script

component, in which the system listens for one of the following keywords: "joke" or "picture". By default, there is typically a third exit, apart from the two exits corresponding to the two above keywords, the third exit being "not found", i.e. neither of the two keywords is identified within 5 seconds (user-defined "listen and sense" time).

Fig. 7 is a display screen showing characteristics of a TALK/PLAY script component, in which Johnny tells a joke. The script component of Fig. 7 includes the identity of the speaker, the source of the desired audio output (text rather than wavefile), and so on.

Fig. 8 is a display screen showing characteristics of a DISPLAY GRAPHICS script component.

Fig. 9 is a display screen which is shown when a script creator selects TEXT.

Fig. 10 is a display screen showing fields via which properties of a selected END script component may be defined.

Fig. 11 is a display screen showing fields via which properties of a selected RECORD script component may be defined.

Fig. 12 is a display screen showing fields via which properties of a selected TALK script component may be defined.

Fig. 13 is a display screen showing fields via which properties of a selected LISTEN AND SENSE script component may be defined.

Fig. 14 is a display screen showing fields via which properties of a selected MOVE script component may be defined.

Fig. 15 is a display screen showing fields via which properties of a selected LOCAL MEMORY script component may be defined.

Fig. 16 is a display screen showing fields via which properties of a selected GLOBAL MEMORY (STORAGE) script component may be defined.

Fig. 17 is a display screen showing fields via which properties of a selected CALCULATE script component may be defined.

Fig. 18 is a display screen showing fields via which properties of a selected WAIT script component may be defined.

Fig. 19 is a display screen showing fields via which properties of a selected TIME MARKER script component may be defined.

Fig. 20 is a display screen showing fields via which properties of a selected BOOLEAN (CONDITION) script component may be defined.

Fig. 21 is a display screen showing fields via which properties of a selected EXECUTE script component may be defined.

Fig. 22 is a display screen showing fields via which properties of a selected GOTO (JUMP) script component may be defined.

Fig. 23 is a display screen showing fields via which properties of a selected CALL SCRIPT script component may be defined.

Fig. 24 is a display screen showing fields via which properties of a selected RANDOM script component may be defined.

Fig. 25 is a display screen showing fields via which properties of a selected INTERNET script component may be defined.

Fig. 26 is a display screen showing fields via which properties of a selected GRAPHICS script component may be defined.

Fig. 27 is a display screen showing fields via which a script creator may define properties of a SCHEDULER option, selected by means of a Scheduler control button in control button array 20. As shown, the properties of the scheduler typically comprise a selection of at least one script, identified by file name, to be executed and a corresponding selection of dates and times at which the script/s are to be implemented.

Fig. 28 is a display screen of a Project Manager option which may for example be selected by pushing a manager button located hierarchically below the Project command in the command bar 10. The Project Manager provides a list of all scripts and means of adding new scripts and removing old ones.

Fig. 29 is a display screen which is typically generated during run time of a script, responsive to a command from a script creator who wishes to monitor the operation of the toys executing the script. The display screen of Fig. 29 preferably includes at least the following information:

- a. The volume of the verbal interaction or sound;
- b. An indication of words spotted by the speech recognition function and the level of confidence thereof;
- c. A list of memory fields with their values.

Figs. 30A - 30B, taken together, form a flowchart illustration of a main script for operating a toy called Storyteller, constructed and operative in accordance with a preferred embodiment of the present invention, the flowchart illustration being suitable for display in

the work area of the display screen of Fig. 1. The main script of Figs. 30A - 30B calls three other scripts, namely Game, Song and Story.

A description of the main script illustrated in Figs. 30A - 30B is as follows, where each paragraph is a description of an individual object:

Start1(Start): Starting point for execution.

CurrentTime(TimeMarker): Sets memory cell <CurrentTime> with the current date and time.

Memory():

Condition1(Condition): Follow the true branch if the value of memory cell <CURRENTTIME> is greater than "04:00:00 PM", or the false branch, otherwise.

Condition2(Condition): Follow the true branch if the value of memory cell <CURRENTTIME> is smaller than "11:59:00 AM", or the false branch, otherwise.

Talk_Evening(Talk): Play WAV file F:\Creator\storyteller\Audio\Op025m.wav using the computer's sound card.

Talk_morning(Talk): Play WAV file F:\Creator\storyteller\Audio\Op015m.wav using the computer's sound card.

Talk_afternoon(Talk): Play WAV file F:\Creator\storyteller\Audio\Op020m.wav using the computer's sound card.

ListenAndSense1(ListenAndSense): Listens for one of the keywords (rainbow,bubble gum) for 5 seconds and put the result in memory cell <SET SECRET NAME>.

Welcome rainbow(Talk): Play WAV file F:\Creator\storyteller\Audio\Op045m.wav using the computer's sound card.

Welcome bubble gum(Talk): Play WAV file F:\Creator\storyteller\Audio\Op050m.wav using the computer's sound card.

Welcome default(Talk): Play WAV file F:\Creator\storyteller\Audio\Op036m.wav using the computer's sound card.

ListenAndSense2(ListenAndSense): Listens for one of the keywords (story,game,song) for 5 seconds.

Talk7(Talk): Play WAV file F:\Creator\storyteller\Audio\Op060.wav using the computer's sound card.

Script_story(Script): Runs the "C:\Serial IDE Storyteller\StoryTeller\Scripts\story.script" script.

Script_game(Script): Runs the "C:\Serial IDE Storyteller\StoryTeller\Scripts\story.game" script.

Script_song(Script): Runs the "C:\Serial IDE Storyteller\StoryTeller\Scripts\song.script" script.

ListenAndSense3(ListenAndSense): Listens for one of the keywords (story,game,song) for 5 seconds.

Wait1(Wait): Does nothing.

Storage1(Storage): Sets global storage cell "secret name" to the same value as memory cell <SET SECRET NAME>.

Memory1(Memory): Sets memory cell <SET SECRET NAME> to "rainbow".

Condition secret name(Condition): Follow the true branch if the value of memory cell <SET SECRET NAME> is equal to "rainbow", or the false branch, otherwise.

Jump1(Jump): Execution continues at node "Welcome rainbow".

Jump2(Jump): Execution continues at node "Welcome bubble gum".

Wait2(Wait): Does nothing.

Wait3(Wait): Does nothing.

Figs. 31A - 31B, taken together, form a flowchart illustration of the Game script called by the main script of Figs. 30A - 30B, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1.

Figs. 32A - 32B, taken together, form a flowchart illustration of the Song script called by the main script of Figs. 30A - 30B, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1.

Figs. 33A - 33B, taken together, form a flowchart illustration of the Story script called by the main script of Figs. 30A - 30B, the flowchart illustration being suitable for display in the work area of the display screen of Fig. 1.

A preferred method for using the system of the present invention to create a script and to subsequently manipulate toys in accordance with that script, is now described.

a. Develop computer controlled toys having talking capacity and/or motion capacity and/or listening capacity. Preferred computer controlled toys and techniques for making toys are described in the Gabai et. al. Patent documents mentioned in the Background section and also in the following U.S. Patent Applications to Gabai et. al.: 09/081,889; 09/062,502;

09/062,547; 09/062,500; 09/062,499; 09/062,579; and 09/081,255.

Operatively associate the toys, initially with a workstation (for the purposes of steps (c) and (d) below) and subsequently with a suitable personal computer such as a home personal computer (for the purposes of step (e) below).

b. Define the above toys using a workstation which may also comprise a personal computer.

c. Use the workstation to generate a script file and executable file representing a desired script to be executed by the personal computer in conjunction with the toys.

d. While the toys are operatively associated with the workstation, run the script file on the workstation to test whether the toys are programmed as desired.

e. While the toys are operatively associated with the personal computer, run the executable file on the personal computer to manipulate the toys in accordance with the desired script.

A preferred method for performing step (b) is as follows:

Battery, Identify, Record, and Help Icons

Several icons appear at the top of the System Test screen: battery icon (under certain conditions), toy icon, base station icon, Record button, and Help icon.

Battery icon: Appears and flashes when the base station detects that a Living Toy's batteries are low.

Toy icon: Searches for a Living Toy. When a toy is not found, a red "X" appears over the icon. When a toy is found, the toy's number is entered automatically in the Toy Number field and the system detects and notes the communications channel in use by the toy.

To search for a toy and its channel:

1. Delete the value that appears in the Toy Number field.
2. Click on the Toy icon.
3. The PC automatically searches for a connected toy. When it finds a toy, the toy emits a beeping noise and the toy's number appears in the Toy Number field. The system registers the RF channel on which the toy is communicating. The system is now able to communicate with the toy.
4. After the system locates a toy, the red "X" that covered the Toy icon disappears.

5. To find another toy, repeat from Step 1.

Base station icon: Searches for a connected base station. When a base station is not found, a red "X" appears over the icon. After the system locates the base station, the red "X" that covered the Base Station goes away.

Record button: Opens the default recording program that resides on your computer. You use this program to record and save WAV files for use in your system test activities. When you are recording, make sure you set the recording properties of the program as follows: 8 KHz ; m-Law or 16 bits ; mono

Toy Number: Enter here the number assigned to the toy you are using. (You will find the number on the back of the toy.) OR: Delete the value that appears in this field and click on the Toy icon. The system will perform an automatic search for the toy number and the RF channel at which it is communicating, and will enter this toy number in the Toy Number field. See Toy icon, above.

Test icons

Three test icons appear at the top left of the System Test screen: a toy icon, a PC icon, and a Stop sign icon.

Toy icon: Runs the selected function through the toy.

PC icon: Runs the selected function through the PC.

Stop sign icon: Stops function that is currently running.

Function buttons

A column of buttons appear along the left side of the System Test screen. You use these buttons to perform the different functions that make up the system test.

Talk: Tests the toy's ability to speak. You need to select a WAV file or enter text in the

Text-to-Speech box, fill in the other values on the screen, and run the test through the toy or the PC.

Listen: Tests the toy's ability to recognize an answer. You need to specify the words you want the toy to recognize, run the test through the toy or PC, state the words, and view the results.

Dialog: Tests the toy's ability to talk, wait for an answer, and recognize the answer. You need to specify what the toy will say and recognize, run the test through the toy or PC, state the words, and view the results.

Move: Tests the toy's ability to move. You need to specify the kind of movement you want the toy to make and run the test through the toy or PC.

Sense: Tests the toy's sensors. You need to define the source of output for the test (the toy or the simulation), press on a sensor on the toy itself or on an onscreen simulation of the toy, and view the results.

Record: Enables the toy to record the user and then tests the recording. You need to specify the name of the file, record the speech, and test the recording by playing it back via the toy or PC.

Hardware: Tests various hardware components that make up your Living Toy system.

Script: Lets you build and test a script, using graphical objects.

Setting: Lets you modify and add descriptions to the sensor and movement lists. You then use your customized list when assigning sensors and/or movements to the toy, in the Talk, Dialog, Move, and Sense pages.

Other: Several features at the bottom of the screen assist you during the testing process.

Events window (long, white box at bottom of screen):

Displays ongoing chronology of events during each test. Refer to this window to see if a test is successful. Note any error messages.

Set Default: Reverts all settings to system default settings.

You: Toy defining user

Exit: Leaves the Living Toy System Test and returns to your desktop.

Reference is now made to Figs. 34 - 46 which are typical screen displays generated during toy definition and testing.

Fig. 34 is a screen display of a main toy definition and testing menu.

Fig. 35 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Talk option in the main menu of Fig. 34.

Fig. 36 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Listen option in the main menu of Fig. 34.

Fig. 37 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Dialog option in the main menu of Fig. 34.

Fig. 38 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Move option in the main menu of Fig. 34.

Fig. 39 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Sense option in the main menu of Fig. 34.

Fig. 40 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Record option in the main menu of Fig. 34.

Fig. 41 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Hardware option in the main menu of Fig. 34.

Fig. 42 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the More option in the menu of Fig. 41.

Fig. 43 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Script option in the main menu of Fig. 34.

Fig. 44 is a screen display which is displayed to a toy defining user response to selection by the toy defining user of the Listen screen object in the screen display of Fig. 43.

Fig. 45 is a screen display which is displayed to a toy defining user response to selection by the toy defining user of the Talk screen object in the screen display of Fig. 43.

Fig. 46 is a screen display which is displayed to a toy defining user responsive to selection by the toy defining user of the Setting option in the main menu of Fig. 34.

Reference is now made to Fig. 47 which is a screen display which is displayed to a toy defining user who seeks to combine a plurality of interconnected script elements each representing an action performable by the toy. The system is operative to generate an integrated command, for transmission to the toy, which command combines at least two adjacent or successive script elements. In Fig. 47, the toy defining user has selected, by outlining, two (or more) adjacent script elements, namely Talk (50) and Listen&Sense (60). The user then selects a Combine function (under the Tool menu in the illustrated embodiment), responsive to which the system generates a single command integrating both script elements. It is advantageous to combine a Talk operation, prompting a human player to make an oral response, and the subsequent Listen&Sense operation which is intended to pick up that oral response, because otherwise, the oral response may occur before the Listen&Sense operation has been actuated, due to the time required for the system to complete the Talk operation and subsequently initiate the Listen&Sense operation.

Another example of operations which are advantageously combined is a Talk operation and a motion operation simulating the mechanics of talking, such as a Move Lips operation. These operations are advantageously combined because it is desired that the motion occur substantially simultaneously with the emission of speech.

An example of operations which are advantageously combined although they are not adjacent, is two or more operations to be performed by two or more respective toys, where it is desired that both operations be performed simultaneously. In this type of situation, selection of the two screen objects (operations) to be combined typically occurs by clicking on one of the objects and subsequently pressing SHIFT and clicking on the other object.

Reference is now made to Fig. 48 which is a semi-pictorial semi-block diagram illustration of a script development system constructed and operative in accordance with a preferred embodiment of the present invention in which the toy actuated by the script is a physical toy.

Reference is now made to Fig. 49 which is a semi-pictorial semi-block diagram illustration of a script development system constructed and operative in accordance with a preferred embodiment of the present invention in which the toy actuated by the script is a virtual toy represented on a screen display.

Reference is now made to Fig. 50 which is a pictorial illustration of a script developer running a script. As shown, progress through the script is apparent to the script developer. In Fig. 50, the virtual or physical toy is speaking and therefore the appropriate Talk operation, i.e. the current Talk operation, is highlighted.

Fig. 51 is a screen display of a time scheduler forming part of the system, according to a preferred embodiment of the present invention. The time scheduler optionally schedules an e-mail retrieval script.

Fig. 52 is a screen display of an Internet searcher constructed and operative in accordance with a preferred embodiment of the present invention. The screen display includes a representation of a script which actuates the Internet searcher. The script representation comprises a flowchart having, among other objects, Talk objects. Typically, each Talk object on the screen includes a brief title indicative of the speech content of that Talk object.

For example, the title "Keyword select" would be appropriate for a script element which prompts the user to pronounce a keyword to be searched. The title "Boolean select" would be appropriate for a script element which prompts the user to select a Boolean relationship between one or more keywords which the user has indicated that s/he desires to search. The title "Repeat" would be appropriate for a script element which prompts the user to repeat at oral selection which was not heard by the system. The title "more keywords?" would be appropriate for a script element which prompts the user to decide to continue defining keywords. The Globe screen object in the script of Fig. 52 indicates execution of an Internet search function defined previously or subsequently using the screen display of Fig. 25. The title "state results" would be appropriate for a script element presenting to the user the results of the Internet search.

The following specifications present a preferred implementation of the present invention by way of a user manual. The use manual explains the principles, the features and the use of a software tool for generating verbal content and for controlling toys and other manipulable objects, particularly suited for toys operated by a PC computer, in wireless

communication, by means of a radio base station connected to the PC, with a toy controller embedded inside the toy.

The present specification uses the following terminology:

Living Object: a trademark of hardware and software technology for building computer controlled toys and other manipulable objects, and for the generation of verbal content for their control.

Scriptwriter: a trademark of a software program for the generation of verbal content for the control of toys based on Living Object technology.

Base Station: A radio transceiver connected to the PC providing wireless communication between the PC and the toy controller embedded in the computer controlled toy.

Welcome to Living Object technology!

Toys and other objects based on Living Object technology use the computer, wireless communications, and voice recognition software to speak with their users in a human voice, with human-like personality and intelligence. The toys hold entertaining, personalized dialogs with the child, demonstrating knowledge of the child and his/her likes and recalling past interactive sessions.

The Living Object Scriptwriter is a tool for the young of all ages, to use in creating interactive scripts that give the toys speech and personality. These scripts feature content that includes:

Interactive dialog focused on a variety of content and activities

Personalized data

Historical data with same user

Time-linked content

This booklet will show you how to use the Living Object Scriptwriter to write exciting interactive scripts for your Living Object toy.

Chapter 1: Getting Ready

Making Sure Everything is in Place

To start working with the Living Object Scriptwriter (hereafter, Scriptwriter), your computer needs to be turned on. Make sure your Living Toy is turned on and close by and that your Living Object base station is plugged into the computer. Refer to the Setting Up the Base Station and Toy section, below, if you need assistance.

Setting Up the Base Station and Toy

The following section refers to Fig. 53 showing the Living Object base station and Fig. 54 showing the Living Object Toy.

Plug the base station's computer cable into the serial port in the back of your computer.

Plug the base station's electrical transformer into a nearby electrical socket.

Turn the computer on and wait until the computer is fully operational and the desktop is displayed.

Turn the toy's on/off switch to ON. The toy will emit a few beeps and any moving facial parts will move briefly.

The system is now ready for action. It's time for you to prepare a script and get some dialog going with your toy!

When the toy is on but not active for a while, it automatically switches to Sleep mode. See the section for instructions on how to wake the toy up.

Opening the Living Object Scriptwriter Software

The following section refers to Fig. 55 showing a screen display of the Scriptwriter icon on desktop and Fig. 56 showing a screen display of the Living Object Scriptwriter main Screen.

Click on the Scriptwriter icon on your desktop.

The Living Object Scriptwriter program will open to its main screen. The screen on your computer should look just like the one below:

Telling the System Which Toy You are Using

Reference is now made to Fig. 57 showing the "select tools – options" screen window display and Fig. 58 showing the "toy" screen window display

Click on Tools-Options. The Options window will open. This window has several "tabs": Toys, Hardware, Environment, Volume Settings, Smart Toy, Scripts, and Reports.

Click on the Toys tab to open the Toy window (do this only if the Toy window isn't already displayed).

Check that, in the Toy List, a check mark appears next to the name of your toy. For example, if your toy is "Monster", check that a check mark appears next to Monster. Information related to the Toy Description for Monster will appear to the left.

Click on Insert/Edit to update the system with the details of your particular toy.

Telling the System to Recognize the Base Station and the Toy

Reference is now made to Fig. 59 showing the "hardware" screen window display.

Click on the Hardware tab to display the Hardware window.

Click on the Check button in the "Check base station" section. The system will begin looking for the base station (which should be plugged into the computer and the wall socket). When it finds the base station, the phrase "Base connected." appears in the Report window.

Click on the Check button in the "Check toys" section. The system will begin looking for the toy that you defined in the Telling the System Which Toy You are Using section, above. When it finds the toy, the phrase "Toy connected." appears in the Report window and details about the toy appear in the Search for toy section of the window.

Making Sure the Toy is Awake

When turned on but not actively engaged in talking, listening, or another function, your Living Object is programmed to slip into "Sleep" mode. Upon doing so, the toy emits a single beep. Even with the beep, you may be concentrating so hard on your scriptwriting efforts that you don't notice the toy has switched to Sleep mode until you try to make it talk or record and get no response.

To switch a sleeping toy back to Alert mode, just:

Press one of its sensors, by squeezing its hand, foot, or other body part.

The toy will emit a beep and be back in Alert Mode, ready for action.

Chapter 2: Scriptwriting Basics

It takes only about two minutes to write a simple interactive script. Here we go!

Reference is now made to Fig. 60 showing the "talk icon" screen display, Fig. 61 showing the Scriptwriter main screen display with the added talk object and to Fig. 62

showing the Scriptwriter main screen display with the added talk object connected by a line to the start object.

Adding a Talk Object

1. Click and hold down the Talk icon (it's the first icon in the group to the left of the Scriptwriter desktop).
2. Drag the Talk icon onto the Scriptwriter desktop, underneath the Start object. The Talk icon now appears in a white box. When on the desktop, the icon is called a Talk object.
3. Move the cursor so that it is on top of the Start triangle. The cursor will change to a pencil. Drag the pencil from the Start triangle to the Talk object.
4. A line appears that connects both objects.

Saving your Script

Reference is now made to Figs. 63 showing the screen display of the action toolbar with the save icon and to Fig. 64 showing the screen display for naming and saving the script.

1. It's time to save the work you've done so far. Click on the Save icon on the Actions toolbar (or select Save from the File menu). The Save window appears.

Note: For your Living Object script to run correctly, you must save all related script and wave files in the same directory (it doesn't matter which directory). If, for example, you create a script consisting of 3 script files and 26 wave files, you must be sure to keep all 29 files in the same directory. The directory does not have to be inside the Scriptwriter directory: it can be in any directory on your hard drive.

2. If necessary, click on the down (browse) arrow to get to the directory in which you want to save the file. If necessary, create a new directory in which you will save all files related to the particular script you are working on. It's a good idea to name the directory after the toy, such as "Monster script." If you haven't yet, double-click on the desired director (whether new or old) so that the file will be saved to it.

3. In the File name field, enter a name for your script, such as "script1." The software automatically adds the extension .script. Click Save.

Adding Speech

Reference is now made to Fig. 65 that is a screen window display of a combo box for typing the toy's speech.

Double click on the Talk object. The Talk window opens.

1. In the first field, marked Toy Name, the name of your toy should appear. If it doesn't, click on the down arrow to the right of the field and choose the name of your toy from the list.
2. Click in the TTS box and type the words that you want the toy to say. You can type anything you want. Let's type a question, to create a script sequence that demonstrates the toy's voice recognition ability. Type the question: "What do you feel like doing now, wise guy? When my eyes light up, say: A joke, a song, or a game."
3. Great stuff! Your toy now has a line of speech that you have created. To hear the voice segment right now, through the toy, click the Play button on the screen—it's the blue button with the green arrow. The toy will vocalize the text you typed, speaking in computer synthesized speech (called "text-to-speech"). Note that you can select text-to-speech that sounds like a man, woman, boy, or girl.

To hear the toy vocalize the text in a human voice, you will need to record the line. See the Recording Speech section, below.

Recording Speech

Reference is now made to Fig. 66 that is a screen window display for recording sound to be played by the toy.

Note: You can record the toy's speech through the toy or through your computer's microphone. Logistically, it may be easier to use a conventional microphone. If you are recording through the toy, make sure it is on and awake. See the section for assistance. If you are using the computer's mike, make sure it is plugged into the microphone jack in the back of the computer and that the speakers and sound software are on. Consult your speaker or sound software manuals for instructions, if necessary.

1. In the Talk window, click on the Record button: it's the blue button with the red circle. The Sound Recorder window opens.
2. As soon as you click on the microphone icon, you are recording. Speaking clearly and holding your mouth close to the microphone, record the line, "What do you feel like doing now, wise guy? When my eyes light up, say: A joke, a song, or a game." When you're done, click again on the microphone icon.
3. Play back your recording to make sure it recorded well. Click on the speaker icon (it's to the left of the microphone icon).
4. If you are not satisfied with the recording, do it again. If you want to increase or decrease the volume of the recording, adjust the Volume dial by twisting it to the left or right with the mouse. Then record the line again.

Saving a Recording

Reference is now made to Fig. 67 that is a screen window display for saving the recording.

1. You need to save the recording. Click on the Save icon. The Record into window appears. You need to save the recording in the same directory as your Scriptwriter script.
2. Click on the down arrow to get to the necessary directory (remember, it's the directory in which you saved your script file earlier, in the "saving your script" section.) Then save the recorded file under any name you choose, such as "wav1." The software automatically adds the extension .wav. Once saved, the recording becomes a sound file, which is also known in the recording industry as a "wave" file.

Playing Back a Recording (through the toy)

Reference is now made to Fig. 68 that is a screen window display for selecting a "wave" file to be played by the toy.

Now we can hear a line that you created and recorded, played through your toy.

1. If the Talk window isn't already open, double-click on the Talk object on the desktop.
2. In the Talk window, click on the circle next to WAV. This tells the system that you want to play a pre-recorded wave file rather than the text-to-speech version of the text that appears in the TTS box.
3. Click on the open file icon (to the right of the WAV field) and browse until you find the wave file you just recorded and saved. Click on the Open button to select the file. The file and its directory path now appear in the WAV field.
4. Click on the Play button (it's that blue square with the green arrow to the right of the WAV field). The system plays the wave file through the toy.

Note: No sound comes out of the toy? Relax. Your toy may have gone to sleep while you were busy scriptwriting and recording. To wake the toy up, squeeze its hand or another body part that contains a sensor. If the toy responds with movement and/or a beep, then you have switched it back to Alert mode. Now repeat Step 4 above.

Still didn't work? Perform one or more of the following checks:

1. Make sure that the name in the Toy Name field at the top of the window is that of your toy (if "Computer" appears, then make sure you change the name to your toy name. See the Adding Speech section, above.
2. Make sure the WAV circle is selected, rather than the TTS circle. This time you want the toy to vocalize the wave file rather than the computer synthesized (TTS) version of the text.

Make sure you follow the system set-up steps for the base station and toy in the "Telling the System to Recognize the Base Station and Toy" section, above.

Adding Listening

Reference is now made to Fig. 69 that shows the Listen icon and to Fig. 70 that is a screen display of a part of the Scriptwriter main window with the Listen object added.

Adding the Listen object to your script

1. Click and hold down the Listen icon (it's to the right of the Talk icon).
2. Drag the Listen icon onto the Scriptwriter desktop, underneath the Talk object. The Listen icon now appears inside a white box. When in this form, on your Scriptwriter desktop, the icon is a Listen object.
3. Move the cursor over the Talk object until it changes to a pencil. Then drag a line from the Talk object to the Listen object. The script will now flow from the Start object to the Talk object to the Listen object.

Now let's tell the toy what to listen for.

Defining Keywords

Reference is now made to Fig. 71 that is the "Listen and Sense" screen window display.

In the previous section, you added an object that tells the toy to listen. Now let's tell the toy what words to listen for. In defining the Talk object, we told the toy to tell the user: "...say: a joke, a song, or a game." Each of these phrases is a keyword phrase that we will now define.

1. Double-click on the Listen object. The Listen and Sense window opens. In the Listen and Sense window, you define what words the toy will listen for or what sensors will be in input mode during the current listen and sense segment.

2. Double check that the correct name appears in the Toy Name field. Click in the Keywords field.
3. Type the keywords, following the same spacing and punctuation pattern you see in parentheses. Type: a joke, a song, or a game.
4. If you wanted to make one or more of the toy's sensors active at this point, you would click the sensor number that corresponds to each of the sensors.
5. Click OK. Part of the list of keywords appears on the listen icon, as a point of reference.

Note: To improve the accuracy of the keyword recognition, try to use keywords that have a least two syllables and make sure that the keywords in a particular group of keywords sound different from each other. You can use keyword phrases that consist of two words.

Sometimes the system doesn't know how to pronounce a keyword. This typically happens when you use special names or made up words. No problem! Just click on the play button to hear how the system reads out the word. Then adjust the spelling of the word and play the word again, and repeat this process as necessary, until the system pronounces the word correctly.

Creating a Response for Each Keyword

Reference is now made to Figs. 72 that is the "Keyword link box" in the "Choose Link" screen display and to Fig. 73 showing the Scriptwriter main screen display with a Listen object links to corresponding Talk objects.

The toy needs to give a different answer to each keyword it hears. This process of building questions, keywords, and responses to keywords gives the toy its intelligent conversational ability--at the most basic level. The system offers many different features that enable you to give the dialog a highly intelligent aspect, such as random answers, answers based on memory, and answers based on collected personal data.

To create a response, simply add a Talk object for each keyword (see the "Adding Speech" section).

Let's add a response to each of the keywords that we have already created:

1. Drag the Talk icon over to the Scriptwriter desktop and under the Listen object. Connect the Listen object to the Talk object. The keyword link box appears, with the first keyword in the list that you entered in the Listen window.
2. Click on OK. (If this is not the right keyword, just click the down arrow and scroll down until the correct keyword appears. Then click OK.)
3. Drag the Talk icon over to the desktop four separate times, until you have four Talk objects beneath the Listen object. Connect a keyword link to each Talk object, as in the previous step.
4. Now let's type some kind of verbal answer for each Talk object. Double-click on the first Talk object, which links to the "a joke" keyword. In the TTS box, type: "You want to hear a joke? You must be a very funny person."

Note: It's a good idea to repeat the keyword at the very beginning of the toy's response. This tells the user that the toy indeed understood the spoken keyword.

5. For each keyword, type an appropriate response in the TTS box of the corresponding Talk object.

Note: You may notice that the fourth keyword automatically displays a link called "Not-Found." This link allows you to create a verbal response to a situation in which the toy did not hear or understand any of the keywords it was listening for (or, if the toy was awaiting sensor input, did not feel input to the sensor that was waiting for input). For instructions on creating a "Not-Found " reaction by the toy, see the section below.

Creating an Answer to Not-Found

Sometimes the system doesn't understand the user's response (or the user didn't provide a response at all). The fourth Talk object that you created needs to contain speech that tells the user what to do if the toy didn't understand or hear the keyword spoken by the user. Typically, you will want to ask the user to repeat the keyword or make a comment to the effect that the toy did not get one of the expected answers and is therefore moving on to the next point in the

script. If you ask the user to repeat the keyword, you should remind the user what the keywords are, just in case she or he has forgotten them.

In the fourth Talk object, which is linked to a "not found" situation:

1. Let's type text that tells the user to repeat the keyword. Double-click on the Talk object. In the TTS box, type: "I'm sorry, but I didn't quite hear what you said. Please tell me again. Say: a joke, a song, or a game."
2. OR: Let's type text that tells the user that the toy did not hear the response, but is moving on to the next point in the script. In the TTS box, type: "Hmmm, you want me to choose? Ok, I'm in the mood for a joke!"
3. You now need to link the objects accordingly. If you typed the text in Step 1, then you need to draw a link from the fourth Talk object back to the Listen object immediately above. If you typed the text in Step 2, then you need to draw a link from the fourth Talk object to the Talk object that provides a response to the keyword "joke."

Running a Script

Reference is now made to Fig. 74 showing the Run-Run screen window display and to Fig. 75 showing the Sample error message screen window display.

You now have enough of a script to run a talk-listen-respond sequence through the toy. Let's give it a whirl!

1. Make sure the toy is awake by squeezing one of its sensors.
2. Select Run-Run.

Note: To run the script from a certain point rather than the beginning, simply click on the object from which you want to run the script and select Run-Run from Selected.

3. The Start icon on your desktop will become highlighted and the Living Object software will run through the script, highlighting each icon as that part of the script is activated.

4. If there are any problems with the script, a window with error messages will appear, like the one below:

The errors listed indicate a problem with Talk1 and Talk5. These errors were generated when the Run-Run option was selected and the toy was still in Sleep mode. The system found an error with Talk1 because it was the first segment of the script that the system could not execute. The error in Talk5 reflects the inability of the sleeping toy to listen at all.

5. Ideally, as you run the script, your toy will voice the texts you defined in Talk1, listen for one of the three keywords you defined in Listen1, and respond accordingly by voicing the text from Talk2, or Talk3, or Talk4.

Note: You will no doubt notice that Talk1 is a wave file, whereas the other Talk objects are played through the toy as synthesized speech. To run the entire script as wave files—that is, in natural voice—you need to open the Talk window of each of the Talk objects and record the text, as you did in the Recording Speech section, above.

Chapter 3: Functions

This chapter serves as a reference guide to each of the functions that appears on the Scriptwriter main screen.

Reference is now made to Figs. 76A and 76B, taken together, show a table of the functions provided by the Scriptwriter with their icons as presented on the Scriptwriter main screen display.

Talk Object

Reference is now made to Fig. 77 showing the Talk screen window display.

To enter into the talk options window double click on the icon on the script.

Click on the Advanced button for the following additional options :

- Toy Name: Determined according to toy available (appears in all of the motion group options)
- Name: Name of object (appears in all of the motion group options)

Note : Toy name and name options appear in all of the motion group objects

- TTS (Text to Speech) field: enter in text to be spoken by toy
- You can change the type of voice used by clicking on the different options available of the right of the TTS field e.g. man, women, boy, girl.
- You can insert a wav file by choosing the wav option field and allocating a wav file either from you computer or a recorded wav file.
- You can record your own message by selecting the record button .This will bring you to the Sound recorder window (see section of recording speech).
- You can also playback your wav file from these window by clicking on the play button.
- Movement Options allow you to select the type of movement for the talk segment.
- The Mood and Stage field are used for additional comment information.

Listen & Sense Object

Reference is now made to Fig. 78 showing the Listen and Sense screen window display.

- Toy Name: Determined according to toy available
- Name of object
- The keywords field is where you define the options available for speech recognition. With the say keywords button which is located at the end of the keywords field you can hear the words chosen.
- The Sensors field allows to define the sensor areas located on the toy for non verbal response.
- Listen time allows you to define the maximum time given to listen or wait for activation of sensors.
- The Memory field allows you to save the results of the recognition process.

- In order to change the accuracy level click on the “Active” field and then “ok” This will bring you to the Script Properties window. Here you can change the speech recognition accuracy level. Please note that the lower the level of accuracy the more sensitive the recognition is.

Move Object

Reference is now made to Fig. 79 showing the Move screen window display.

- The Movement field allows you to pick the type of activity the Toy will make.
- The Moving Time field defines the length of time the movement will take place.
- When you choose the “Run in the Background” option you are instructing the toy to continue with the script once receiving the movement command.

Record Options

Reference is now made to Fig. 80 showing the Record Options screen window display.

The Record option allows you to record your voice, your friends or anyone else.

Wav File Name – Insert name of file that will be recorded in the script.

Memory Object

Reference is now made to Fig. 81 showing the Memory screen window display.

Allows you to put a certain compartment of the computer's memory and give the compartment a name.

Condition Object

Reference is now made to Fig. 82 showing the Condition screen window display.

Compare two different values or check if one value is greater then, less then, equal to, or not equal to a certain value.

Calculation Object

Reference is now made to Fig. 83 showing the Calculation screen window display.

Do some math on the values that are stored on the computer's memory compartments. The computer can add, subtract, multiply and divide.

Random Object

Reference is now made to Fig. 84 showing the Random screen window display.

Create a list of values that the computer will choose from a random basis. Tell the computer in which memory compartment to put the values.

Time Marker Object

Reference is now made to Fig. 85 showing the Date and Time screen window display.

Put a certain time or date in a compartment in the computer's memory.

Wait Object

Reference is now made to Fig. 86 showing the Wait screen window display.

Instruct the toy to wait for a certain amount of time before proceeding with the script.

Jump Object

Reference is now made to Fig. 87 showing the Junp screen window display.

Skip to a different point in the script.

Execute Object

Reference is now made to Fig. 88 showing the Execute screen window display.

Allows you to run and software program on the computer.

Script Object

Reference is now made to Fig. 89 showing the Run Script screen window display.

Enables you to run any other Scriptwriter Script.

Internet Object

Reference is now made to Fig. 90 showing the Internet screen window display.

Opens a defined web page.

Graphics Object

Reference is now made to Fig. 91 showing the Graphics screen window display.

Shows a picture or video file on the computer's screen.

Image (Advanced Options)

- Display time is the length of time the image/video will be shown

- Size field allows you to determine the height and width of the image chosen.
- Choose Display - this function is used for limiting and controlling the display panels.

Video (Advanced Options)

When choosing the "Wait until finish" command this instructs the toy to wait until the video is completed before continuing with the script.

Reference is now made to Figs. 92, 93, 94, 95 and 96 showing the corresponding "End", "Script Properties", "Choose Link", "Popup Menu" and the "Options" screen window displays.

End Object

The end object stops the script and allows for the users to define the exit names. When opening the script from a different window and the single output mode is not defined the user will be able to view all the available script exists.

Chapter 4: Artificial Life

Once you've written a script, the IDE lets you activate it in a way that gives life-like behavior to the toy. The IDE comprises algorithms and a very strong compiler that integrate time, pattern, and minimal interval and apply them to the script or a collection of scripts. The resulting artificially created life for the toy is so authentic, users can easily forget they are speaking and interacting with a toy. This section describes how to use the IDE to create artificial life.

Artificial life is divided into three main screens, the Editor, the Manager, and the Viewer.

Each of these screens is presented in a separate section of this chapter.

Artificial Life Editor

There are two kinds of AL editors: Professional and non-Professional.

Professional Editor

Reference is now made to Fig. 97 showing the Artificial Life Algorithm Editor screen window display.

The Artificial Life Professional Editor allows you to define formulas and assign values to local and system parameters that will later act on a given script. You use the Editor to write your own formulas or edit pre-written formulas provided by the function library. The Editor then allows you to create an algorithm from the formula you have defined, and associate the algorithm with the current script.

In the current example, a formula and parameters are being defined to determine how often the script entitled Games.script. will be executed.

In the Behavior parameter box, four parameters must be assigned values: memory, initial Priority, threshold Priority, and minimum Interval.

To do a test run on the algorithm, you need to assign the formula a temporary value in the Formula parameter box. For example, the formula on the sample screen has been assigned a value of 1. This value could represent successful completion of, say, a lesson in science. If the script has never been completed successfully, it could have formula parameter value of 2.

Editor

Reference is now made to Fig. 98 showing the Artificial Life Editor screen window display.

The editor enables to build formula for specific script. Here the steps to add AL formula to script.

First you have to choose the script by pressing load button. Then fill in formula by double click on a cell. You must fill in at least one cell. Finally you have to save the AL formula by pressing the save button.

Cell management

Reference is now made to Fig. 99 showing the Artificial Life Editor screen window display with the Cell Management popup window.

By pressing on the right click on the mouse, you get a popup with cell functions.

Function Library

Reference is now made to Fig. 100 showing the Artificial Life Editor screen window display with the Function Library popup window.

Add specific function. Select the function then fill in properties. Press OK in the properties section or double-click on the selected function to add the function, or press esc button to cancel.

Artificial Life Manager

Reference is now made to Fig. 101 showing the Artificial Life Manager screen window display.

The Artificial Life Manager gives you a clear and precise overview of all scripts that the Artificial Life engine will check and the formulas, parameters, and values assigned to them. You can work from the Manager to make changes to the definitions. The Manager contains functions for adding, removing, and viewing the history of the 10 last executions each script. Highlighting a script name with the highlight bar displays all the relevant details of that script.

Artificial Life Viewer

Reference is now made to Fig. 102 showing the Artificial Life Editor Viewer window display.

The Artificial Life Viewer presents a real-time and historical graphical depiction of the status of up to five scripts at any one time. You use the Viewer to track the behavior of different scripts, as determined by the value stored for each script in "Memory." You can select the "Show activation level" item to view the threshold of the selected scripts, and thereby determine when the last time was that each script executed. The Viewer displays the last 10 minutes of Artificial Life activity on an ongoing basis, scrolling to the right as additional charted activity takes place.

Building Artificial Life Environment

Building artificial life environment is could be complicated. Here are advises how to build AL toy.

1. Make list of scripts.
2. Make list of parameters.
3. Make Tables: Dependence table and formula table.
4. Fill in formula table.
5. Fill in dependence table.
6. Build the scripts.
7. Register/Add the scripts.

Examples of dependence table and formula tables.

Dependence table

| parameter -> script | eat | laugh | drink | game | song |
|------------------------|-----|-------|-------|------|------|
| eat | | | | | |
| laugh | | | | | |
| drink | | | | | |
| Game | | | | | |
| Song | | | | | |

Formula table

| Script | formula | threshold | priority | description |
|--------|---------|-----------|----------|-------------|
| Eat | | | | |
| Laugh | | | | |
| Drink | | | | |
| Game | | | | |
| Song | | | | |

Examples of AL scripts

Reference is now made to Figs. 103 and 104 showing the Scriptwriter main screen display with corresponding game script and laugh script.

Function Bar Commands

File Menu

Reference is now made to Fig. 105 showing the Scriptwriter main screen display with the File menu open.

The file menu allows you to create a new script, open an existing script and other operations that are found in a normal file menu.

New Script

In order to begin writing your script, click on new script in your file menu and a new window will appear on your screen. Now you can begin working on your script.

Open Script

To open an already saved script, click on open script in your file menu. A window will open up containing a list of the existing scripts you can search from. When you find the script you are looking for, click on its name, for example script1.script, and the script file will open.

Download Script

The download script command in your File Menu opens a collection of existing scripts from Creators' internet site. You can download an existing script from the web to your IDE Scriptwriter program.

Save Script

To save a script you have created on the program click on the Save Script command in your File Menu.

Save Script As

To save your new script under a certain file name, click the Save Script As command on your File Menu. A window will open up asking for your scripts' name, name your script, press the save command and your file is now saved in the directory you have assigned to it.

Save Script As Image

The Save Script As Image command saves your script in the format of a picture image. The script is saved as a Metafile Image (WMF). WMF is especially compatible with the Word program. When you save your script in the form of WMF you can make corrections and changes, outside the IDE Scriptwriter program, in Word itself.

Create Report

Reference is now made to Fig. 106 showing the Scriptwriter main screen display with the Create Report Window.

The Create Report command creates a chart in the Excel program which documents which objects appear in the script you have created. In the window that opens, when you click on Create Report, You can chose to chart all properties of all existing objects by pressing Print All .

You can limit the chart to a specific object, for example talk, by selecting Create Selected in the window that opened when Create Report was clicked on.

Print Description

When clicking on the Print Description command a detailed text file and NOT a chart appear. The same information, which appears in the Create Report chart, appears in Print Description in textual form.

Print Preview

When clicking on the Print Preview command you will receive a print preview of the script you have just created.

Print

The Print command will print a visual picture of your script as well as a verbal description of the stages of your script.

Below the print command in your file menu appear the Last Opened Scripts. It can display a maximum of the last Three files that have been worked on.

The last command in your File Menu is the Exit command. When clicking on the exit command you will exit the IDE Scriptwriter Program.

Edit Menu

Reference is now made to Fig. 107 showing the Scriptwriter main screen display with the Edit menu open

The Edit Menu allows you to amend and change the script you have already created. It includes commands such as cut, copy paste.

Undo

Allows you to undo the last operation that was made on the script that you have been working.

Redo

Allows you to redo an undo operation that you have made.

Cut

Allows you to cut a part of your script and to paste it in another place, or to cut a part of the script in order to remove it.

Copy

Allows you to copy a part of your script and to place the same action copied into another part of the script, thus having two operations repeat themselves in two separate parts of the script.

Paste

The paste and copy commands are interconnected. When cutting a part of your script you must then click on the paste command in order to place that operation in another part of your script.

Select All

Allows you to select all parts in your script so that changes and corrections you wish to make can take place in the whole script itself.

Find

Reference is now made to Fig. 108 showing the Scriptwriter main screen display with the Find Window.

Using the Find command you can search for a specific word or object in your whole script, making your search easier.

Replace

Reference is now made to Fig. 109 showing the Scriptwriter main screen display with the Replace Window

When clicking on the Replace a window will appear. This window is split into two sections-Target and Object.

Target- the target defines where the desired replacement should take place. It can take place in a selected part, or in the whole script.

Object- the object defines in which objects the replacement should take place. It can take place in All objects that have shared properties, or you can execute the Replace command according to Object Type. A replacement will be made in a specific object according to its unique properties.

Clipboard

Copy the image or description (Copy Image to Clipboard, Copy Description to Clipboard) of your script onto Windows' clipboard. All Windows applications can now use the image or description.

View Menu

Reference is now made to Fig. 110 showing the Scriptwriter main screen display with the View menu open.

The View Menu offers you different forms of viewing the script you have created, such as zoom in/out, volume etc.

Zoom in

The Zoom in lets you view your script in magnified size.

Zoom out

The Zoom out lets you view your script in decreased size.

Normal Size

The normal Size lets you view your script in its original size.

Volume

Reference is now made to Fig. 111 showing the Scriptwriter main screen display with the Volume and Speech Recognition Windows.

Clicking on the Volume show the volume of all that is spoken or heard in your script. This can help you understand why, for example, words are not being recognized by the program because the microphone level is too low.

SR Result

Speech Recognition (SR) window ,show the speech recognition results during running the script. The accuracy help you determine if the sensitivity in identifying certain parts in the program should be lowered. The higher the % of accuracy the closer the annunciation is to the computers'.

The Rec. Wav button, allows you to hear the last saved recordings during listen process.

Reference is now made to Fig. 112 showing the Scriptwriter main screen display with the Watch List and the Add Memory windows

Watches

Using the Watches command, you can follow the different values of Memory that have been saved, during or after running the script.

Execute log

Reference is now made to Fig. 113 showing the Scriptwriter main screen display with the Execute Log and the Messages windows

The Execute Log is a logger of all operations that have been identified and executed. This can be extremely helpful in identifying errors that have been made.

Messages

When clicking on the Messages, a box will come up on screen identifying any errors that might have been made or any hints the program has for you. If nothing appears in the box no error was found and no hint offered.

Sensor Simulation

Reference is now made to Fig. 114 showing the Scriptwriter main screen display with the Sensor Selection window.

This is a simulation for the sensors of the specific object in your script. The sensors that are active in different parts during the script are identified by name during this Sensor Simulation.

Link Style

This refers to the different styles of links that can be made between two objects in the script (ex. between talk & move). There are six different styles of links for ex. Vertical-horizontal, horizontal-vertical. These different styles help you to better organize your script writing form. Note: You can also change link style by double clicking on the link line itself, in your script.

Simulator

When clicking on the Simulator, a window will open up on your screen. You will see a simulator doll that actually Acts Out your script, only if it is running in simulation mode.

Scheduler

Reference is now made to Fig. 115 showing the Scheduler screen window display.

The Scheduler can determine at what set time you script will be executed. You can schedule your script to appear once an hour, once a day, on an event like a birthday or every time a dolls' hand is pressed. Not only scripts can be scheduled; you can also schedule a Message to be delivered on a set time or event.

You can also receive a List of the last scripts to be run and on what dates their running had occurred.

Scheduler - add task

Reference is now made to Figs. 116 and 117 showing the Scheduler screen window display with the Add Task popup window Scheduler List popup window correspondingly.

Find Toys

Reference is now made to Fig. 118 showing the Scriptwriter main screen display with the Find Toys and Assign Channels window.

This command searches for the marked toys. The toys that are defined, are identified. It can also tell you which toys are awake and which are sleeping, and which do not exist.

Run Menu

Reference is now made to Fig. 119 showing the Scriptwriter main screen display with the Run menu open.

This menu allows you Run your finished script, pause and so on.

Run

Play your script.

Run from Selected

Allow you to begin playing your script from a specific chosen point.

Pause

Pause your script midway at a chosen place.

Stop

Bring your running script to a complete stop.

Check Script

Check your script for any errors or hints (if there are any errors or hints in your script the Message window will appear).

Tools Menu

Your Tools Menu controls all the environment of your IDE scriptwriter program. You can control the toys, volume, sensors etc.

Options

The Options commands are split into 7 pages:

Toys

Reference is now made to Fig. 120 showing the Scriptwriter main screen display with the Option window at the Toys Page.

In the Toys page you can define toys that we show in the list. Toy is define with name- toy name, type- which defines the toy according to the operations it can perform, number, channel. You can also allows you to remove toys from the list.

Hardware

Reference is now made to Fig. 121 showing the Scriptwriter main screen display with the Option window at the Hardware Page.

The Hardware page is split into three subsections.

Check Base Station- checks the communication between the base station and the program, it resets the base station.

Check Toys- the toys which I have chosen to work with are checked.

Search for Toys- searches for a toy according to its number or channel, when the toy is found the program activates the toy which in return makes a sound.

There is also a Report box that reports what has happened, which toy was identified and which wasn't.

Environment

Reference is now made to Fig. 122 showing the Scriptwriter main screen display with the Option window at the Environment Page.

Simulation through the PC- simulation of your script run in the computer.

View simulator- awakens the simulator doll inside the program.

Advance Properties- show every object (for ex. talking, moving) with advanced properties.

Toy Identity- Changes the illustration in the script itself to an illustration of the chosen toy and not the generic illustration. This helps to clarify which toy is talking or operating at different points in a multi-toy script.

Default Toy- the toy in the script will be the default toy.

Volume Setting

Reference is now made to Fig. 123 showing the Scriptwriter main screen display with the Option window at the Volume Setting Page.

These Volume Settings are for the speaker as well as the microphone in the doll. A doll is selected, and the reload button is clicked on. This asks the doll to send the program its' internal volume settings. An update can be made to these settings, saving the update and changing the original settings. After the update, another check of the volume settings is made.

Living Toy

Reference is now made to Fig. 124 showing the Scriptwriter main screen display with the Option window at the Living Toy Page.

In the Living Toy page you can activate all toys that are programmed for artificial life. The toy, like a live person, needs to sleep, eat, wake up at a set time. You can chose to activate only certain toys for artificial life, once you chose them they will "wake up"(if they are sleeping).

Scripts

Reference is now made to Fig. 125 showing the Scriptwriter main screen display with the Option window at the Script page.

When selecting "activate automatic downloading" the scripts from the internet are directly downloaded into a chosen place in your hard disk. This option is only available to those who register. You can chose to receive only scripts that match criteria you have selected.

Report

The Report page has:

Save Logger- every script that has run can be saved along with the date of the running. This can help to keep better track of the scripts.

Save Listen File- can save any listen made in a script (it always saves the last listen heard).

Memory- this allows you to add or remove a memory to the compilation of memories.

Error list

- 1 - SR-Catch all error. Probably an internal error or a subtle corruption of the database.
- 2 - SR-User not found in database.
- 3 - SR-Language not found in database.
- 4 - SR-Syntax not found in database.
- 5 - SR-Context not found in database.
- 6 - SR-Database not found.
- 7 - SR-Dictionary not found in database.
- 8 - SR-Context with this name already exists in database.
- 9 - SR-Language with this name already exists in database.
- 10 - SR-Syntax with this name already exists in database.
- 11 - SR-User with this name already exists in database.
- 12 - SR-Database with this name already exists.
- 13 - SR-Error occurred while trying to activate context on recogniser.
- 14 - SR-Error occurred while trying to activate language on recogniser.
- 15 - SR-Error occurred while trying to activate syntax on recogniser.

- 16 - SR-Error occurred while trying to load user.
- 17 - SR-Grammar load failure.
- 18 - SR-No context defined.
- 19 - SR-No database defined.
- 20 - SR-No algorithm running.
- 21 - SR-No active context.
- 22 - SR-Invalid pointer (in a parameter).
- 23 - SR-Wrong inifile.
- 24 - SR-Access denied.
- 25 - SR-Buffer too small (in a parameter).
- 26 - SR-You cannot perform this action in this state.
- 27 - SR-Could not activate.
- 28 - SR-Out of heap memory.
- 29 - SR-No word recognized.
- 30 - SR-Invalid syntax.
- 31 - SR-Cannot merge given contexts.
- 34 - SR-WORDNOTFOUND-Cannot find or delete word.
- 35 - SR-Word already exists.
- 36 - SR-Class not found in context.
- 37 - SR-Cannot convert BNF file to context.
- 38 - SR-Cannot merge active words.
- 39 - SR-The active context is closed.
- 40 - SR-Cannot open file.
- 41 - SR-Cannot load library.
- 42 - SR-Cannot merge.
- 43 - SR-Wrong type (in a parameter).
- 44 - SR-Unsupported wave format.
- 45 - SR-Already active.
- 46 - SR-Context is still installed.
- 47 - SR-Cannot load context.
- 48 - SR-Context is not active.
- 49 - SR-Cannot load language.

- 50 - SR-Cannot load user.
- 51 - SR-Different languages cannot be active at the same time or trying to compile to a context with a different language.
- 52 - SR-Different users cannot be active at the same time.
- 53 - SR-No wave format specified.
- 54 - SR-Context is active.
- 55 - SR-Language is in use.
- 56 - SR-Language is in use.
- 57 - SR-Cannot create directory.
- 58 - SR-No valid database.
- 59 - SR-Database is opened.
- 60 - SR-Language is already registered.
- 61 - SR-Language is not registered.
- 62 - SR-Context is already registered.
- 63 - SR-Context is not registered.
- 64 - SR-Environment already exists.
- 65 - SR-Environment not found.
- 66 - SR-Cannot delete directory.
- 67 - SR-No dictionary specified.
- 68 - SR-Dictionary already exists.
- 69 - SR-DLL not found.
- 70 - SR-Corrupt DLL.
- 71 - SR-Database is corrupted.
- 72 - SR-Feature is not yet implemented.
- 73 - SR-Invalid input (of a parameter, or input signal).
- 74 - SR-Conversion failed.
- 75 - SR-Unable to copy a file.
- 76 - SR-Unable to delete a file.
- 77 - SR-Context is opened.
- 78 - SR-Bad name.
- 79 - SR-Incompatibility problem.
- 80 - SR-Disk full.

- 81 - SR-Dictionary is opened.
- 82 - SR-Format not found.
- 83 - SR-Symbol already exists in library.
- 84 - SR-Symbol not found in library.
- 85 - SR-Database is in use by a recogniser.
- 86 - SR-Dictionary is in use.
- 87 - SR-Syntax is in use.
- 88 - SR-Error creating file.
- 89 - SR-License Number in asrapi is invalid.
- 90 - SR-No training set found.
- 91 - SR-Property not found.
- 92 - SR-Export not found.
- 93 - SR-Value out of range.
- 94 - SR-No context library defined.
- 95 - SR-Different database used.
- 96 - SR-Error when generating transcription of a word.
- 97 - SR-Agc can not be active during user word training.

- 1 - TTS-File not found.
- 2 - TTS-File creation error.
- 3 - TTS-File writing error.
- 4 - TTS-Memory allocation error.
- 5 - TTS-Memory locking error.
- 6 - TTS-Memory unlocking error.
- 7 - TTS-Memory free error.
- 8 - TTS-Wave Device open error.
- 9 - TTS-Wave device closing error.
- 10 - TTS-Specified waveformat not supported.
- 11 - TTS-No wave devices available.
- 12 - TTS-TTS has not been initialized.
- 13 - TTS-Specified frequency not available.
- 14 - TTS-Specified parameter is out of range.

- 15 - TTS-Specified output PCM format not available.
- 16 - TTS-TTS system is busy.
- 17 - TTS-Not authorized TTS DLL is used.
- 18 - TTS-Dictionary loading error.
- 19 - TTS-wrong dictionary handle.
- 20 - TTS-Wave device writing error.
- 21 - TTS-No input text.
- 22 - TTS-Bad command for current state.
- 23 - TTS-Grapheme to phoneme conversion fail.
- 24 - TTS-Unknown dictionary format has been found.
- 25 - TTS-Creating instance error.
- 26 - TTS-No more TTS instance available.
- 27 - TTS-Invalid TTS instance has been specified.
- 28 - TTS-Invalid TTS engine has been specified.
- 29 - TTS-TTS instance is busy.
- 30 - TTS-TTS engine loading error.
- 31 - TTS-No engine has been selected.
- 32 - TTS-Internal system error.
- 33 - TTS-Specified wave device is busy.
- 34 - TTS-Invalid dictionary entry has been specified.
- 35 - TTS-Too long source or destination text has been used.
- 36 - TTS-Max. dictionary entries are reached.
- 37 - TTS-Specified entry exists already.
- 38 - TTS-Not enough space.
- 39 - TTS-Invalid argument.
- 40 - TTS-Invalid voice id.
- 41 - TTS-No engine has been specified.

{3 digit error x01-x25}

- 1 - Invalid Handle.
- 2 - Device already opened.

- 3 - Device cant setup.
- 4 - Memory allocation.
- 5 - No communication.
- 6 - System.
- 7 - Base not connected
- 8 - Timeout.
- 9 - Invalid register number.
- 10 - Invalid channel.
- 11 - Invalid DeviceID.
- 12 - Wrong state.
- 13 - Invalid parameter.
- 14 - Sound card IN opened.
- 15 - Sound card OUT opened.
- 16 - File open.
- 17 - File create.
- 18 - File read.
- 19 - File write.
- 20 - Format not supported.
- 21 - TTS speech generation.
- 22 - SR engine not active
- 23 - Buffer is too small.
- 24 - SR no active context.
- 25 - TTS engine not active.

It is appreciated that the apparatus of the present invention is useful for generating scripts not only for toys but also for any computer-controllable object.

It is appreciated that the software components of the present invention may, if desired, be implemented in ROM (read-only memory) form. The software components may, generally, be implemented in hardware, if desired, using conventional techniques.

It is appreciated that various features of the invention which are, for clarity, described in the contexts of separate embodiments may also be provided in combination in a single embodiment. Conversely, various features of the invention which are, for brevity, described in

the context of a single embodiment may also be provided separately or in any suitable subcombination.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention is defined only by the claims that follow:

CLAIMS

1. A system for generating scripts having verbal content, the system comprising:
 - a computer having a user input receiver operative to receive a user's definition of a script for at least one computer-controllable animated figure, the script including a plurality of interconnected script elements each representing an action performable by the computer-controllable animated figure, the script comprising at least one verbal script element representing a verbal action performable by the computer-controllable animated figure;
 - a graphics interface operative to generate a pictorial image of the script as the script is generated by the user, the graphics interface including a drag and drop facility operative to drag and drop script elements and a flowchart generating facility, said facilities being operative to provide a flowchart of script elements including illustrated connections, having respective illustrated directions, interconnecting illustrations of each dragged and dropped script element; and
 - a verbal output generator controlled by the user's definition of the script and operative to generate verbal output as defined in the script.
2. Apparatus according to claim 1 wherein said animated figure comprises a virtual animated figure on a computer screen.
3. Apparatus according to claim 1 wherein said animated figure comprises a physical animated figure.
4. Apparatus according to claim 3 wherein the physical animated figure is wirelessly connected to the computer.
5. A script definition method comprising:
 - accepting a user's definition of a speech recognition script element including a designation of a plurality of keywords to be recognized; and
 - providing a flowchart generation user interface operative to facilitate generation of a flowchart by a user representing a script defined by the user, the flowchart including a conditional flowchart element representing the speech recognition script element, the

providing step comprising prompting the user to associate each branch of the conditional flowchart element with one of the plurality of keywords.

6. A method for generating a computerized script including a speech recognition script element, the method comprising:

generating a speech recognition script element including selecting n keywords to be recognized; and

generating a script junction having at least $n + 1$ branches corresponding respectively to the n keywords to be recognized and to a "not found" option which is followed if none of the n keywords are recognized in a user's utterance.

7. A method according to claim 5 or claim 6 and also comprising automatically running the script.

8. A script generating method comprising:

providing a flowchart generation user interface operative to facilitate generation of a flowchart by a user representing a script defined by the user, the script comprising a plurality of script elements represented respectively by a plurality of flowchart elements; and

using a computer to run the script defined by the user including highlighting, for the user, the flowchart element corresponding to the script element currently being executed.

9. A script generation method according to claim 8 wherein the script is defined for an animated figure capable, in conjunction with the computer, of executing each of the script elements and wherein the using step comprises activating the animated figure by running at least a portion of the script on the computer.

10. A system according to claim 1 wherein the computer comprises a simulator operative to simulate at least a portion of the script without utilizing the animated figure.

11. A system according to claim 10 wherein the computer comprises a computer speaker and wherein the simulator is operative to simulate speech of the animated figure by activating the computer speaker.

12. A system according to claim 10 wherein the computer comprises a computer microphone and wherein the simulator is operative to simulate reception of speech by the animated figure by activating the computer microphone.

13. A system according to claim 10 wherein the computer comprises a computer screen and a stored representation of the figure's performance of individual physical-action script elements and wherein the simulator is operative to simulate physical action of the animated figure by displaying, on the computer screen, an image of at least a portion of the figure performing the physical action.

14. A system according to claim 1 and also comprising an Internet searcher.

15. A system according to claim 10 wherein the computer comprises at least one input device and wherein the simulator is operative to simulate sensing activities of the animated figure by receiving mock sensory inputs via the input device.

16. A computerized script generating system comprising:

- a user input receiver operative to receive a user's definition of a script including a plurality of interconnected script elements each representing an action performable by a computer-controllable device; and

- a script implementer receiving the user's definition of the script and operative to generate an integrated command, for transmission to the device, which command combines at least two adjacent script elements.

17. A system according to claim 16 wherein the script implementer is operative, for at least one pair of first and second script elements,

- to generate a single command, for transmission to the device, when said first and second script elements occur adjacently in said script and

- to generate a separate command for each of the first and second script elements when said first and second script elements occur separately rather than adjacently in the script.

18. A system for generating scripts having verbal content, the system comprising:
- a computer having a user input receiver operative to receive a user's definition of a script for at least one computer-controllable physical animated figure, the script including a plurality of interconnected script elements each representing an action performable by the computer-controllable animated figure, the script comprising at least one verbal script element representing a verbal action performable by the computer-controllable animated figure; and
 - a verbal output generator controlled by the user's definition of the script and operative to generate verbal output as defined in the script,
- wherein the computer comprises a simulator operative to simulate at least a portion of the script without utilizing the animated figure.
19. A system according to claim 1 and also comprising a computerized scheduler operative to control the animated figure according to a user-determined schedule received by the scheduler.
20. A system according to claim 19 and also comprising an e-mail function.
21. A system according to claim 20 wherein the e-mail function comprises apparatus for reading e-mail and apparatus for sending e-mail.
22. A system according to claim 1 which is operative to generate a computer executable file operative to manipulate the toy without recourse to generation of scripts.
23. A toy content development system for developing content for interactive computerized toys, the development system comprising:
- for each of at least one computerized toy, a preprogrammed library of software functions which are executable by the toys; and
 - a computing environment within which a user of the system can program at least one of the computerized toys by generating at least one combination of software functions in said library.
24. A system according to claim 23 wherein said combination comprises a logical

combination.

25. A visual toy development system providing a visual environment for developing toy content, the system comprising:

a visual computing environment within which a user of the system can visually program at least one computerized toy by generating at least one combination of software functions, the environment being operative to provide a library of visual software functions which are executable by at least one toy.

26. A system according to claim 25 wherein said environment comprises a preprogrammed library of visual software functions.

27. A verbal interaction development system providing a visual environment for developing a verbal interaction, the system comprising:

a visual computing environment within which a user of the system can visually program a verbal interaction,

the visual computing environment including a visual representation of at least one of a speech comprehension function and a speech generation function.

28. A verbal interaction development system providing a visual environment for developing a verbal interaction, the system comprising:

a visual computing environment within which a user of the system can visually program a verbal interaction,

the visual computing environment including a visual representation of at least one of a speech recognition function and a speech articulation function.

29. A system according to claim 25 wherein the visual computing environment is operative to accept a user's definition of at least one of the following logical structures:

a decision tree;

a flowchart; and

a state diagram,

and to generate a manipulable visual representation thereof;

wherein the visual representation includes a simulation of performance, by the toy, of the combination of software functions generated by the user, the simulation including a traversal of at least one of the logical structures.

30. A verbal interaction development system providing a visual environment for developing a verbal interaction, the system comprising:

- a computing environment within which a user of the system can program a verbal interaction,

- the computing environment including a representation of at least one of a speech recognition function and a speech articulation function,

- the speech recognition function comprising:

- a speech template generator operative to accept, for at least one speech input, a user's definition of a set of templates, to each of which the speech input is to be compared; and

- a template recognition probability evaluator operative to evaluate the probability of correct matching of at least one speech input to each of the templates in the set.

31. A system according to claim 23 and also comprising a developed content simulator operative to generate, on a screen display, a simulation of a programmed computerized toy performing a plurality of software functions in accordance with the programming of the toy by the user, using the computing environment.

32. A system according to claim 23 and also comprising a user workstation serving the user which is located remotely relative to said preprogrammed library and said computing environment.

33. A system according to claim 32 wherein the remotely located user workstation is connected to the preprogrammed library and said computing environment via the Internet.

34. A system according to claim 33 wherein the remotely located user workstation comprises an Internet browser.

1/120

FIG. 1

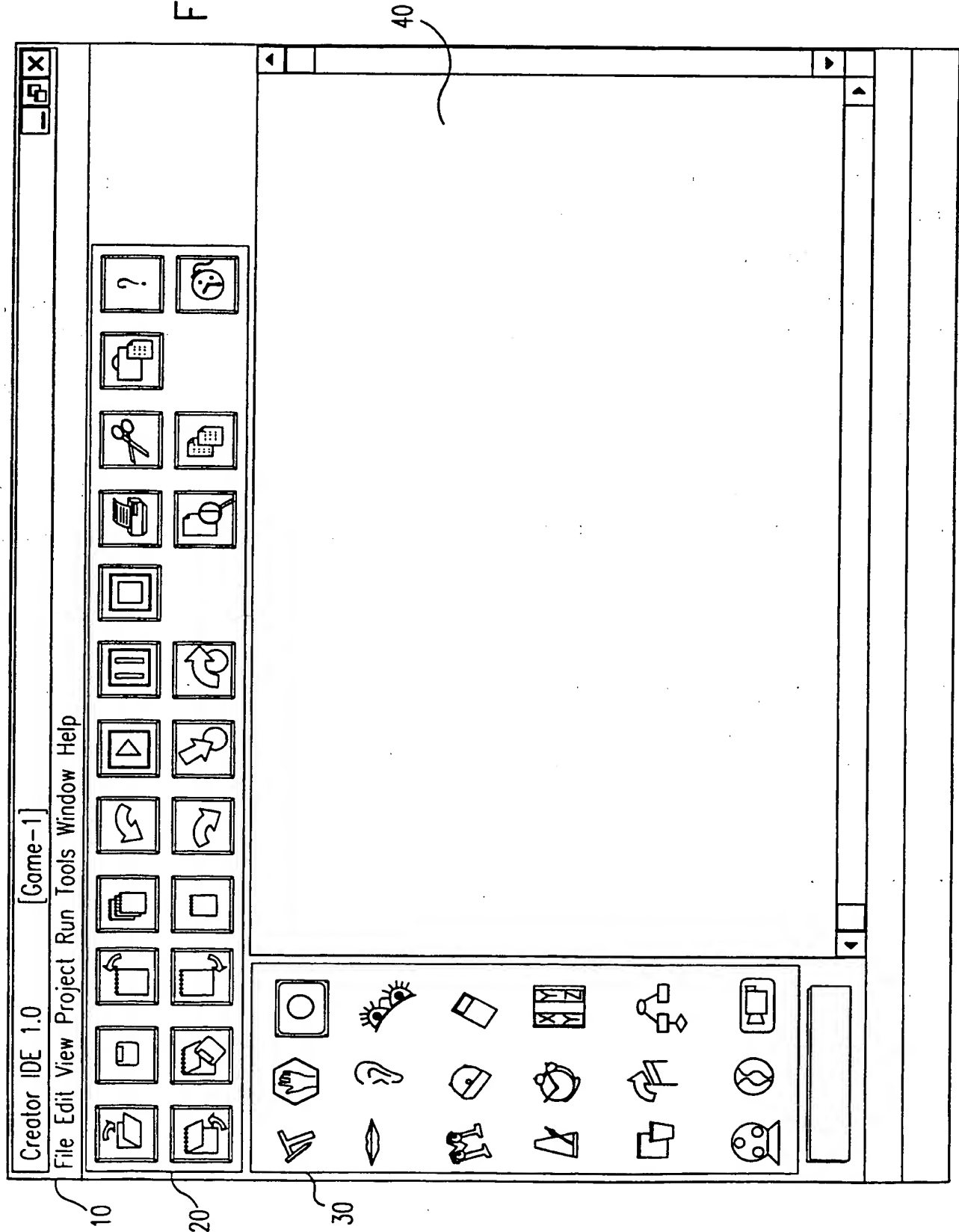


FIG. 2

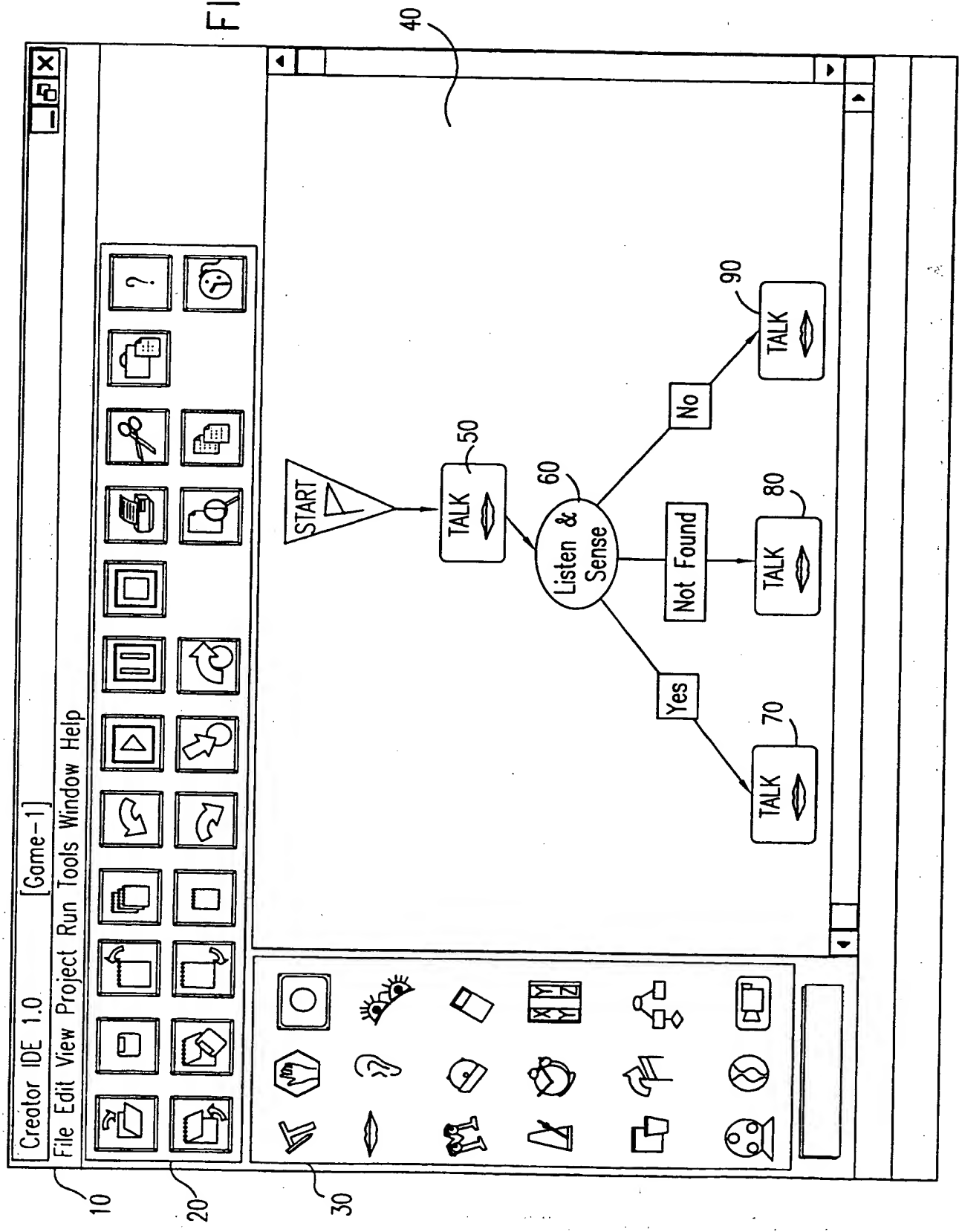
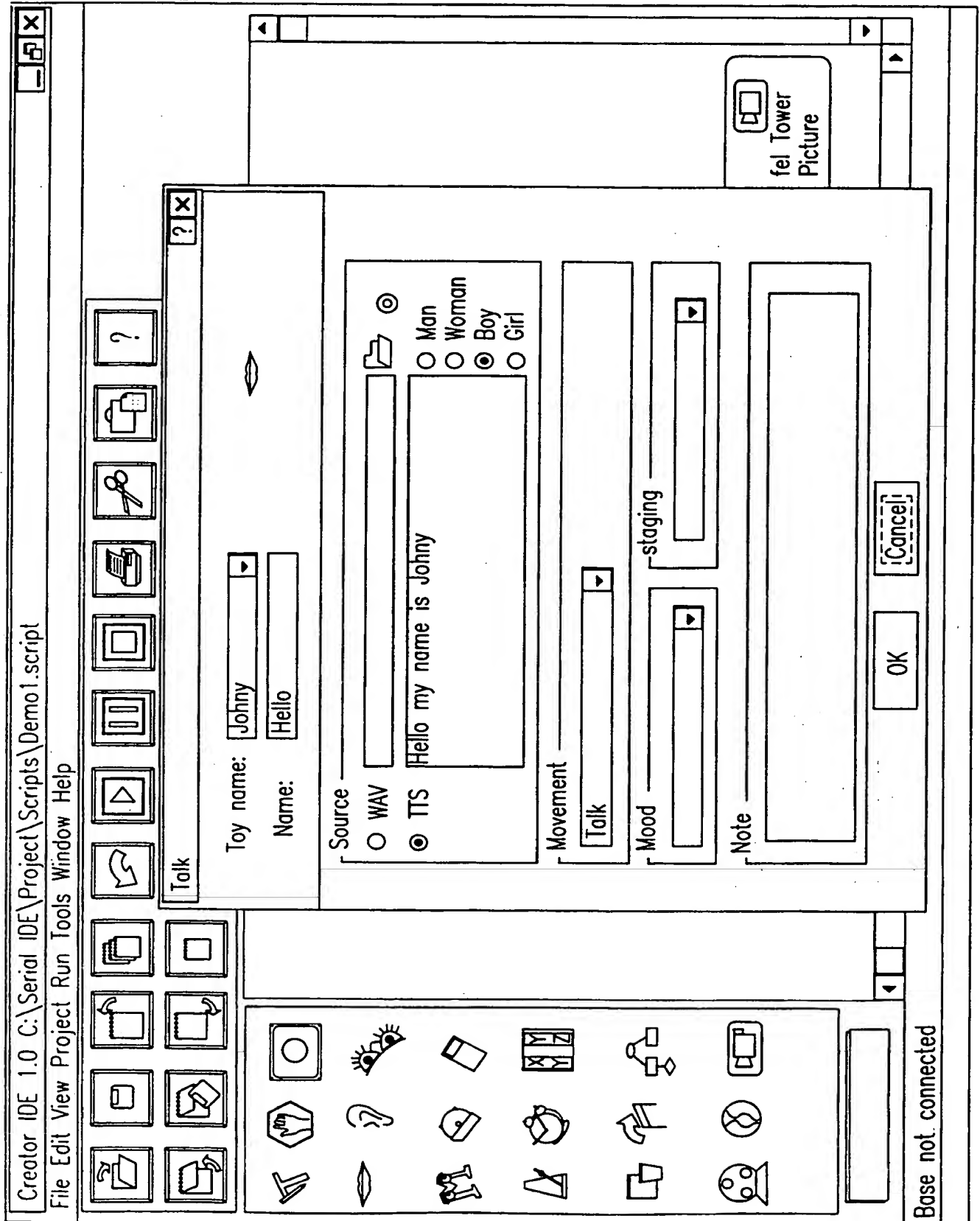


FIG. 3

3/120



4/120

FIG. 4

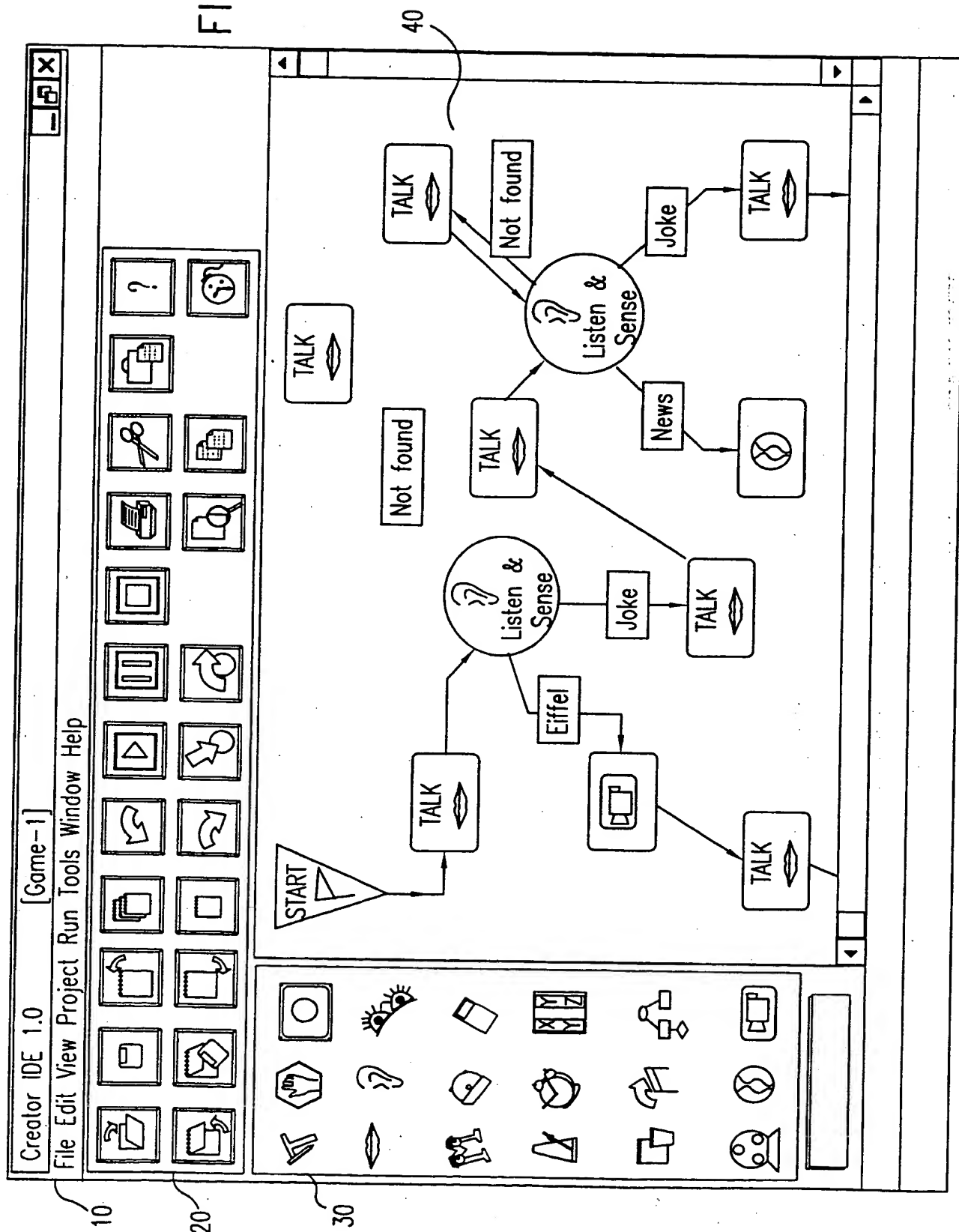




FIG. 5

5/120

Creator IDE 1.0 C:\Serial IDE\Project\Scripts\Demo1.script
File Edit View Project Run Tools Window Help




Talk

Toy name: 


Name:


Source



☐ WAV 


☒ TTS ☐ Do you want to hear a joke or hear the news


☐ Man ☐ Woman ☒ Boy ☐ Girl

Movement 

Talk 

Mood  staging 

Note 

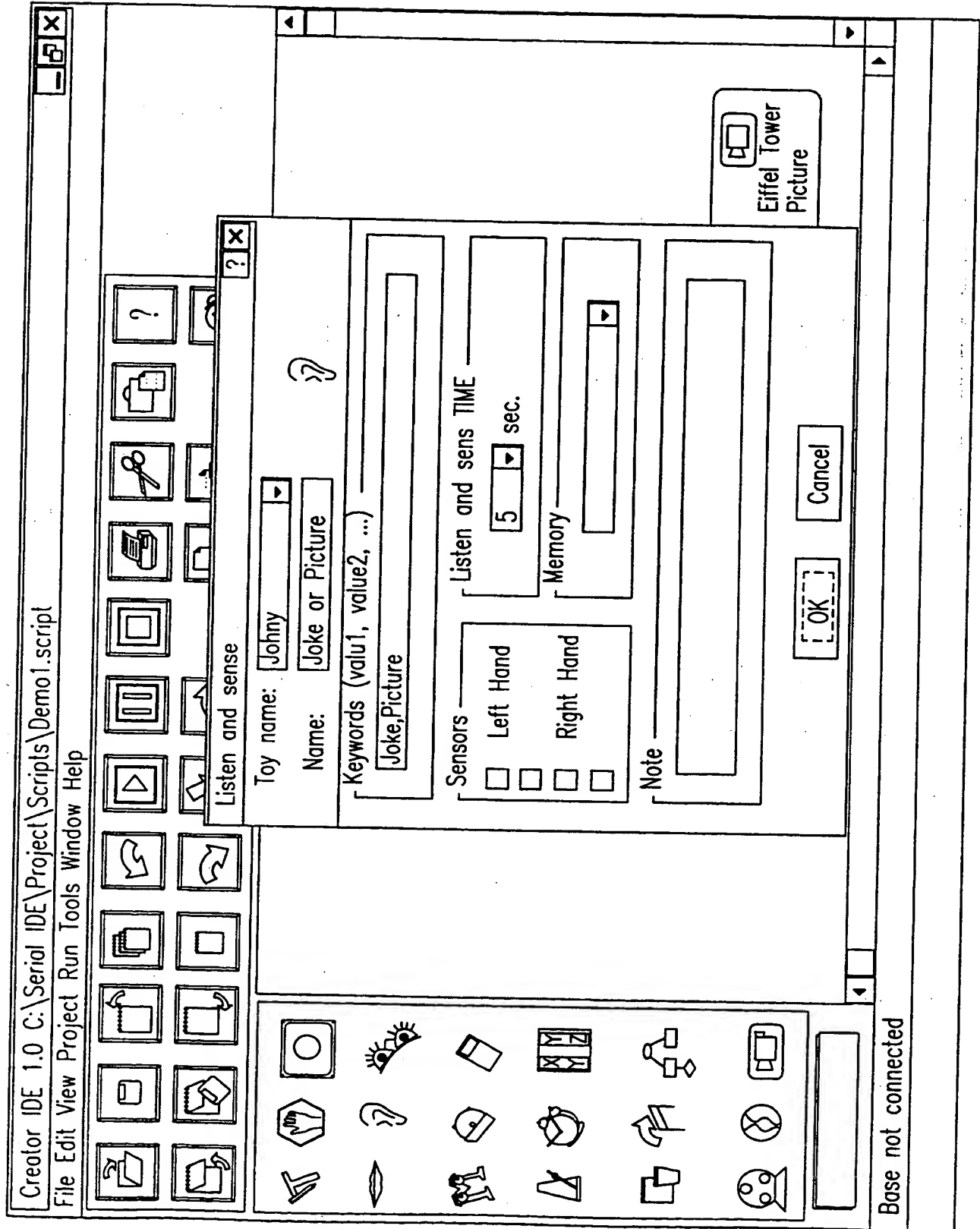
 fel Tower Picture

Base not connected

OK Cancel

6/120

FIG. 6



7/120

FIG. 7

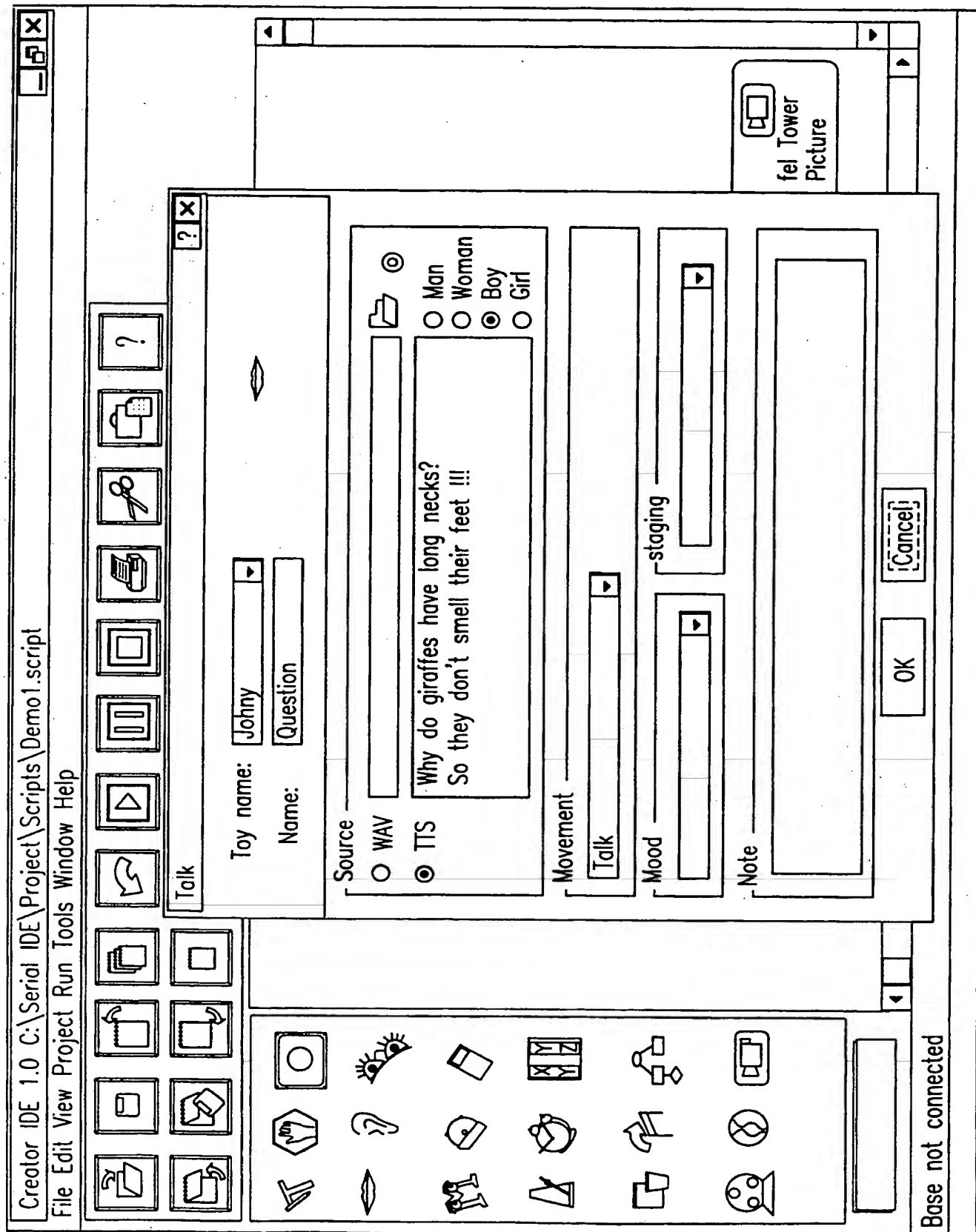


FIG. 8

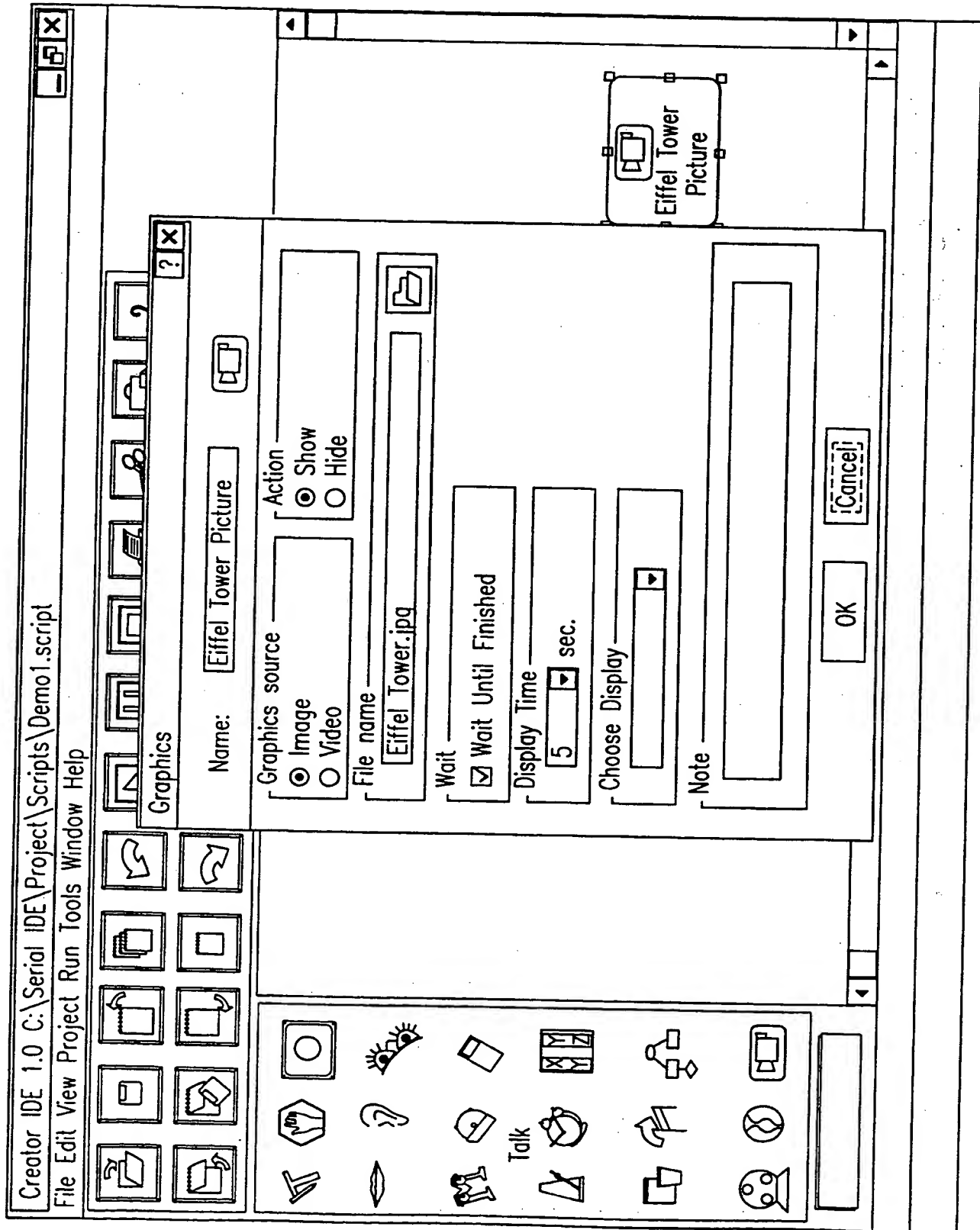
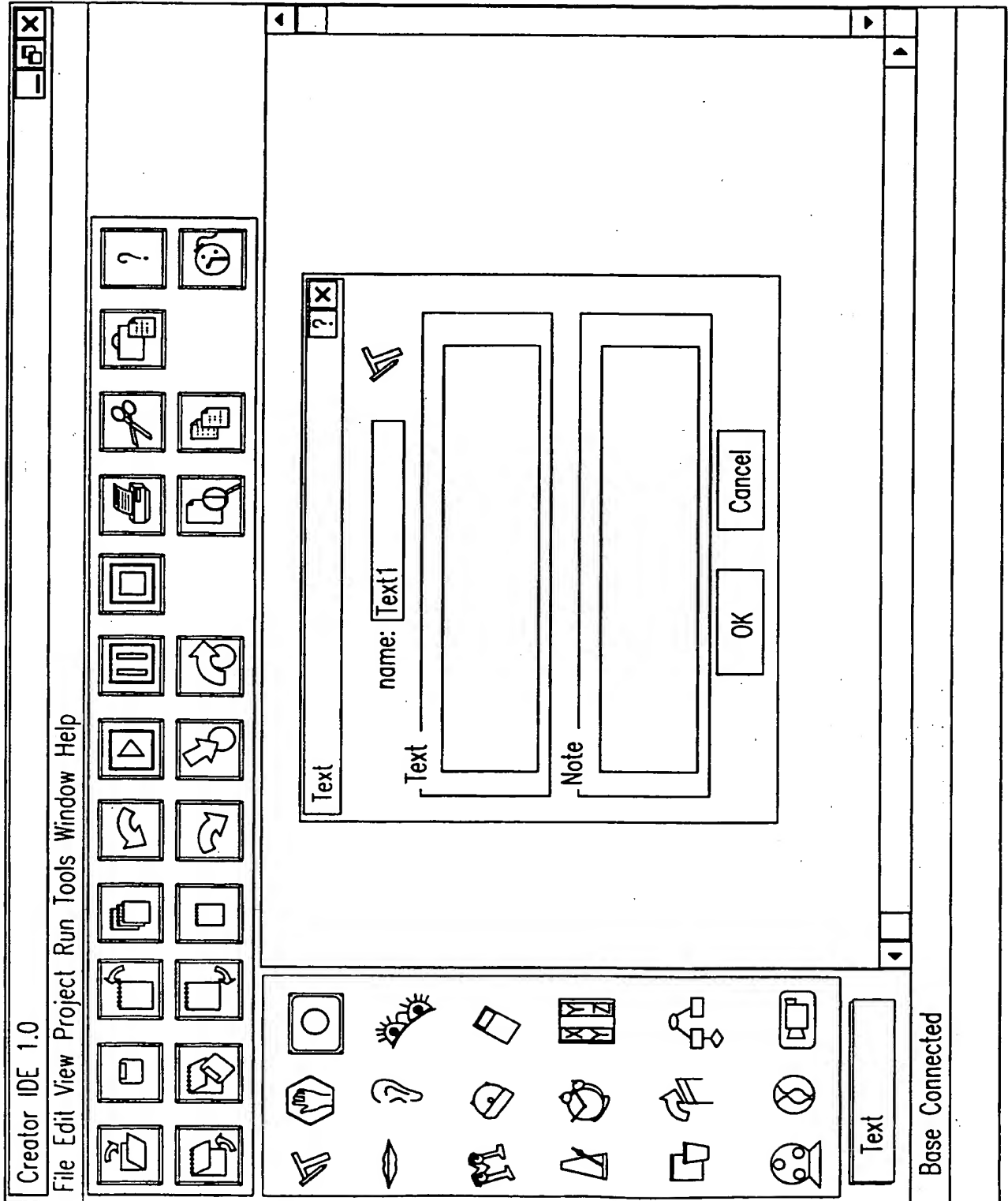
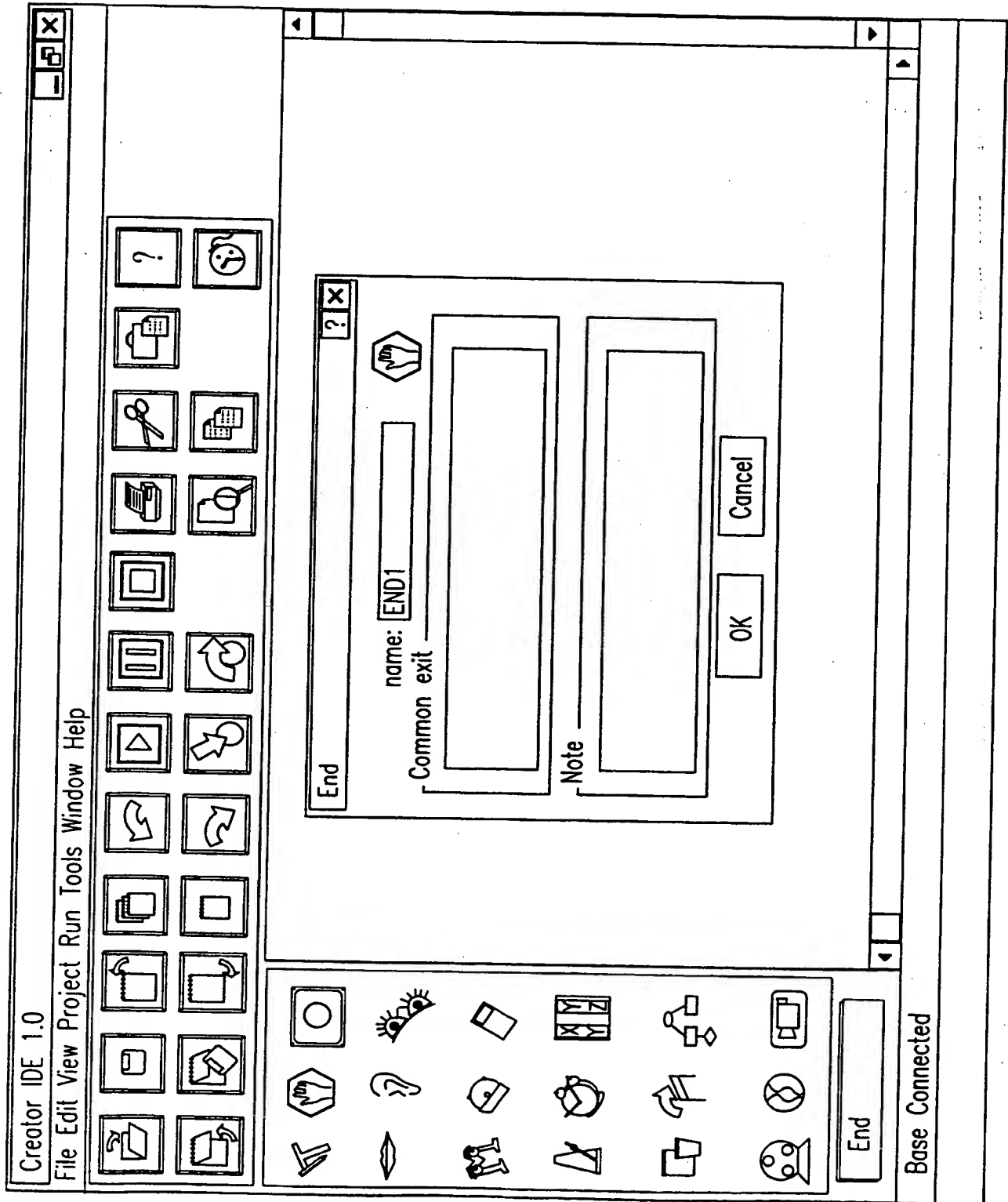


FIG. 9



10/120

FIG. 10



11/120

FIG. 11

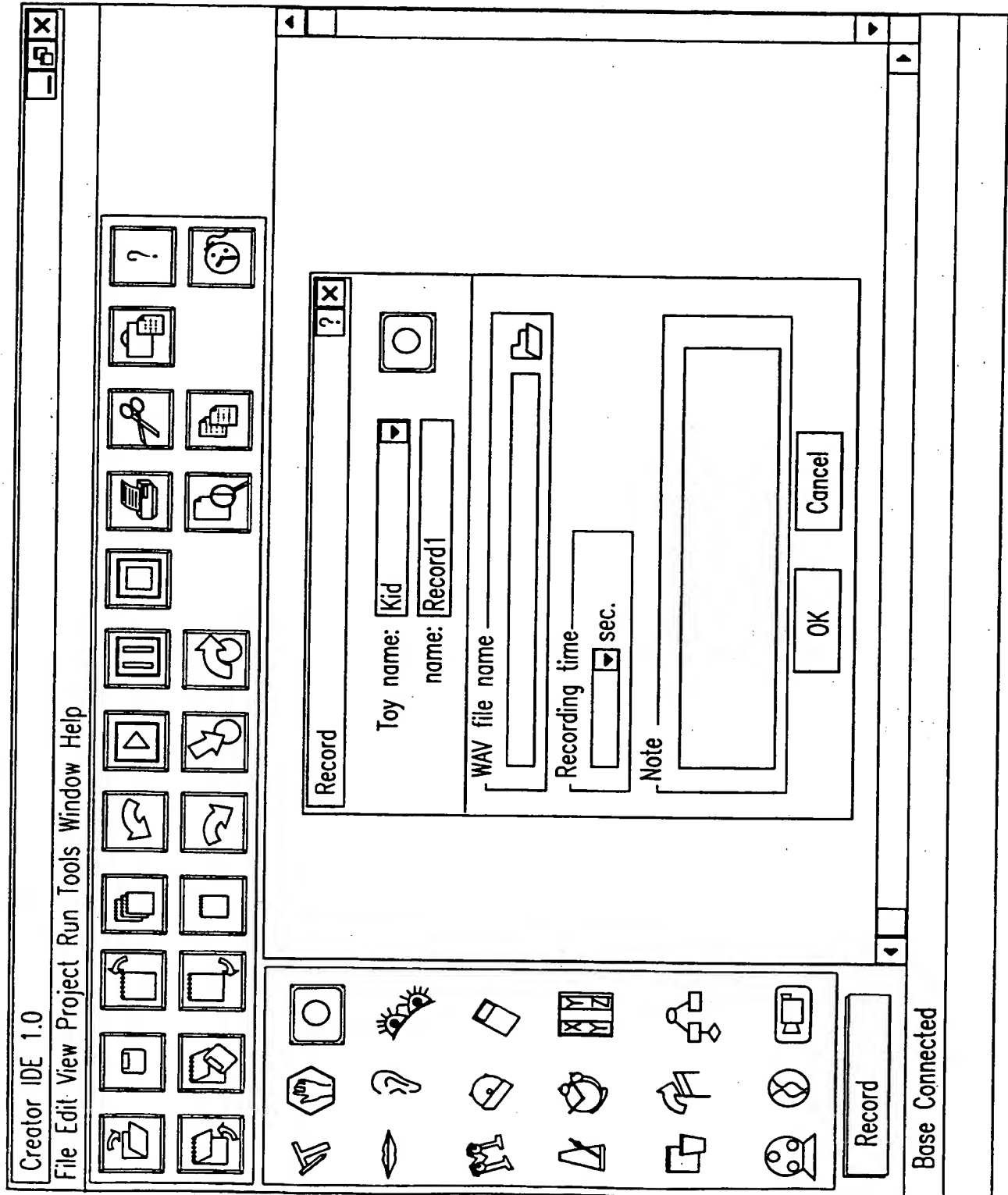


FIG. 12

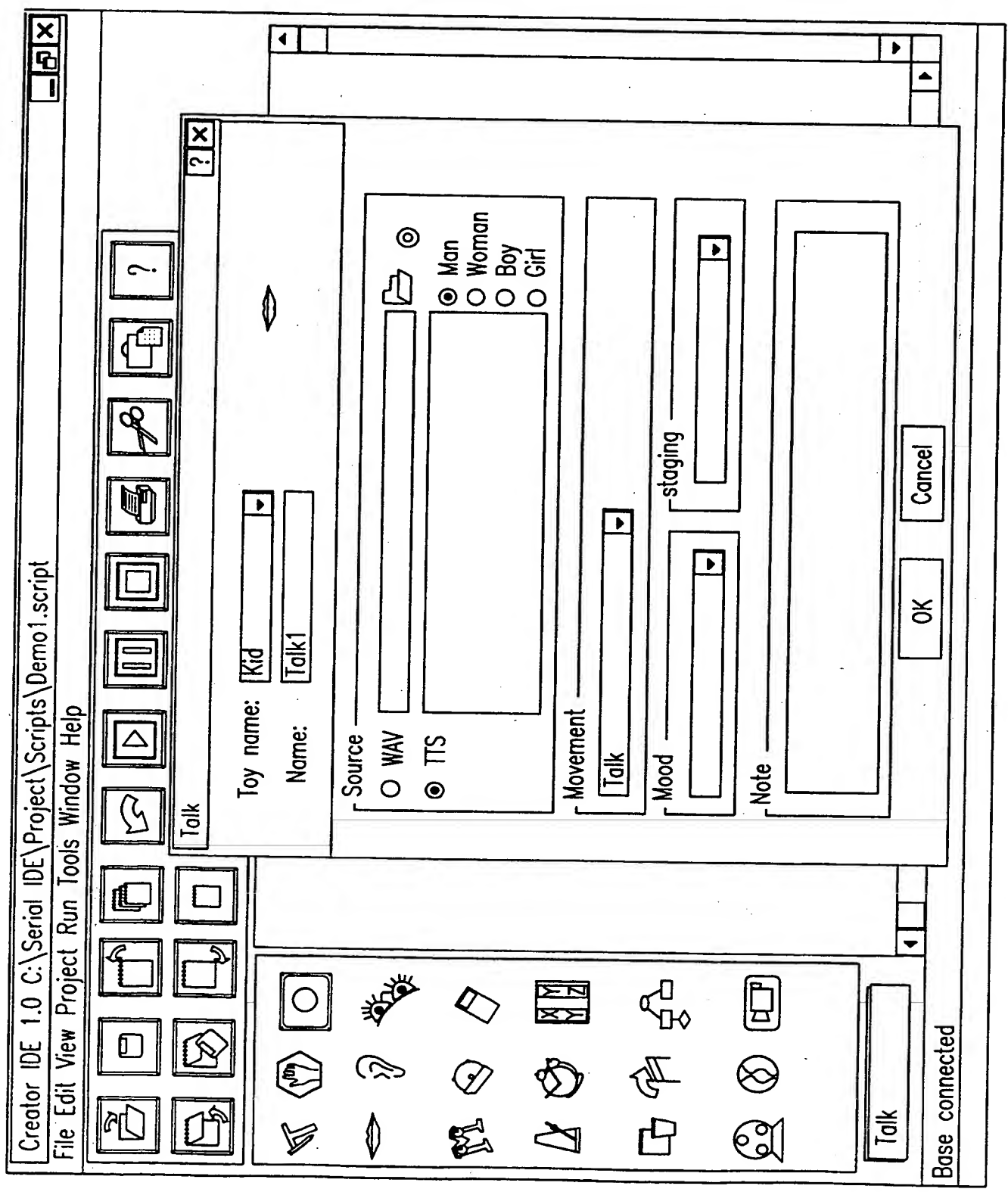
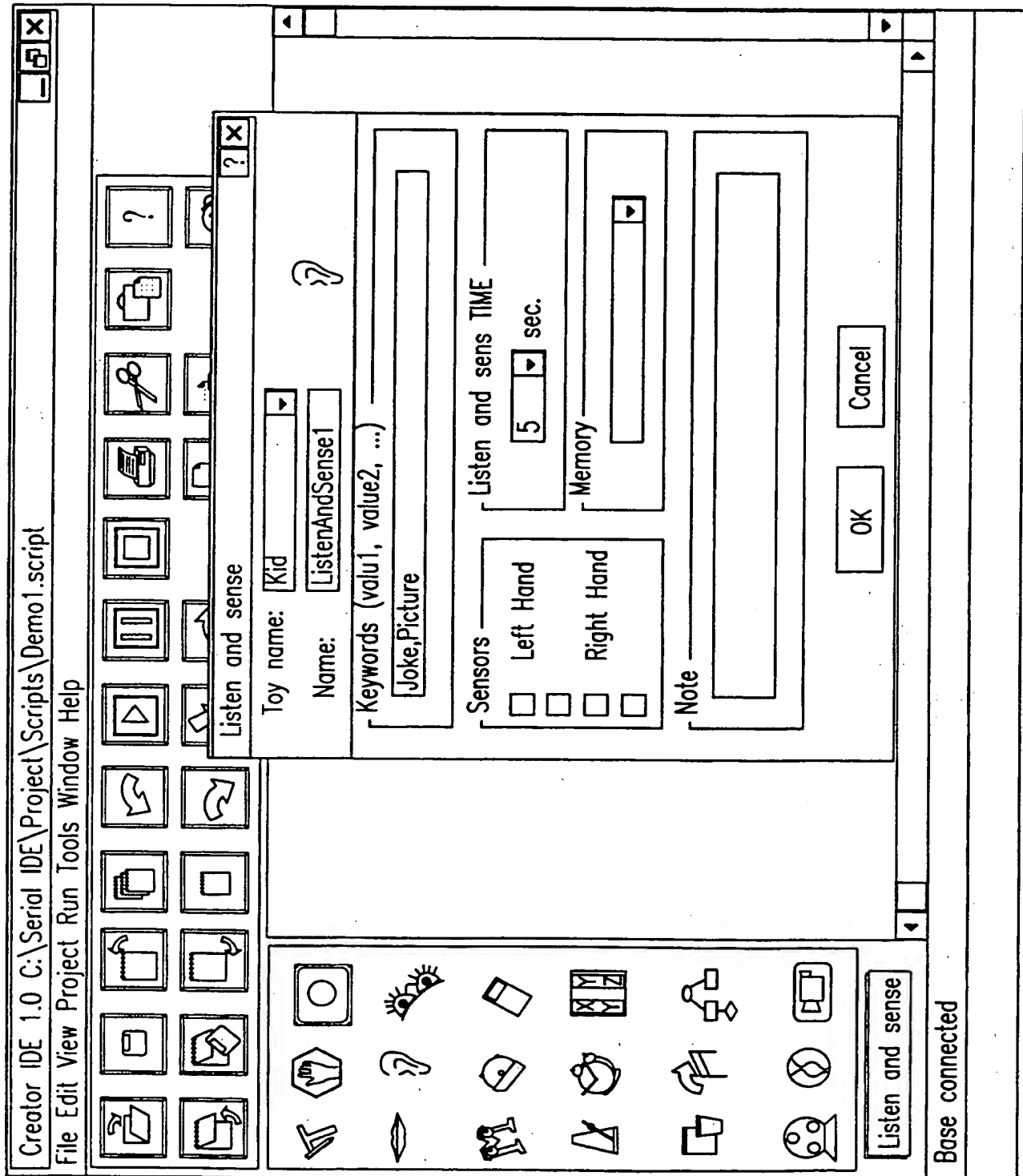
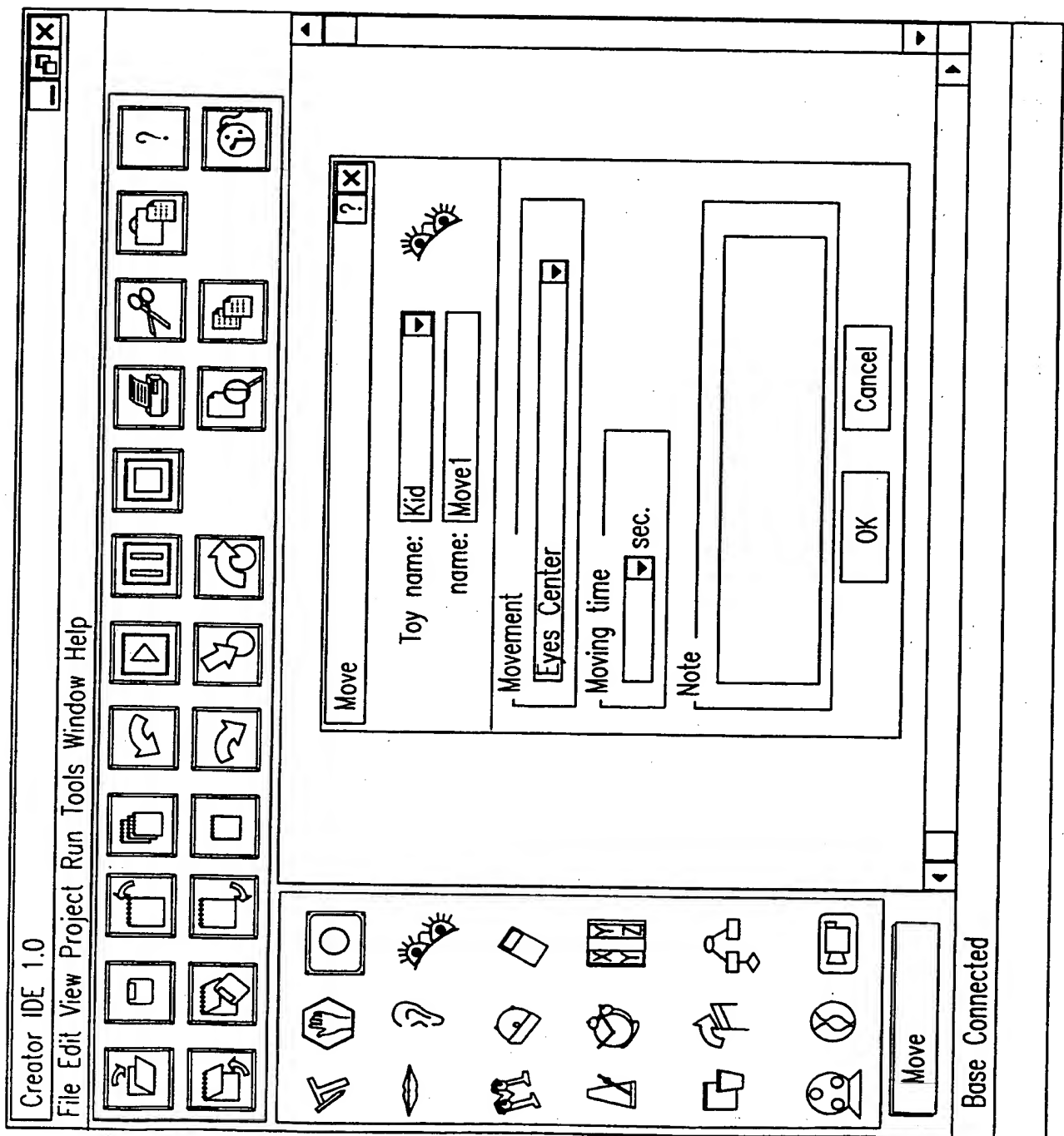


FIG. 13



14/120

FIG. 14



15/120

FIG. 15

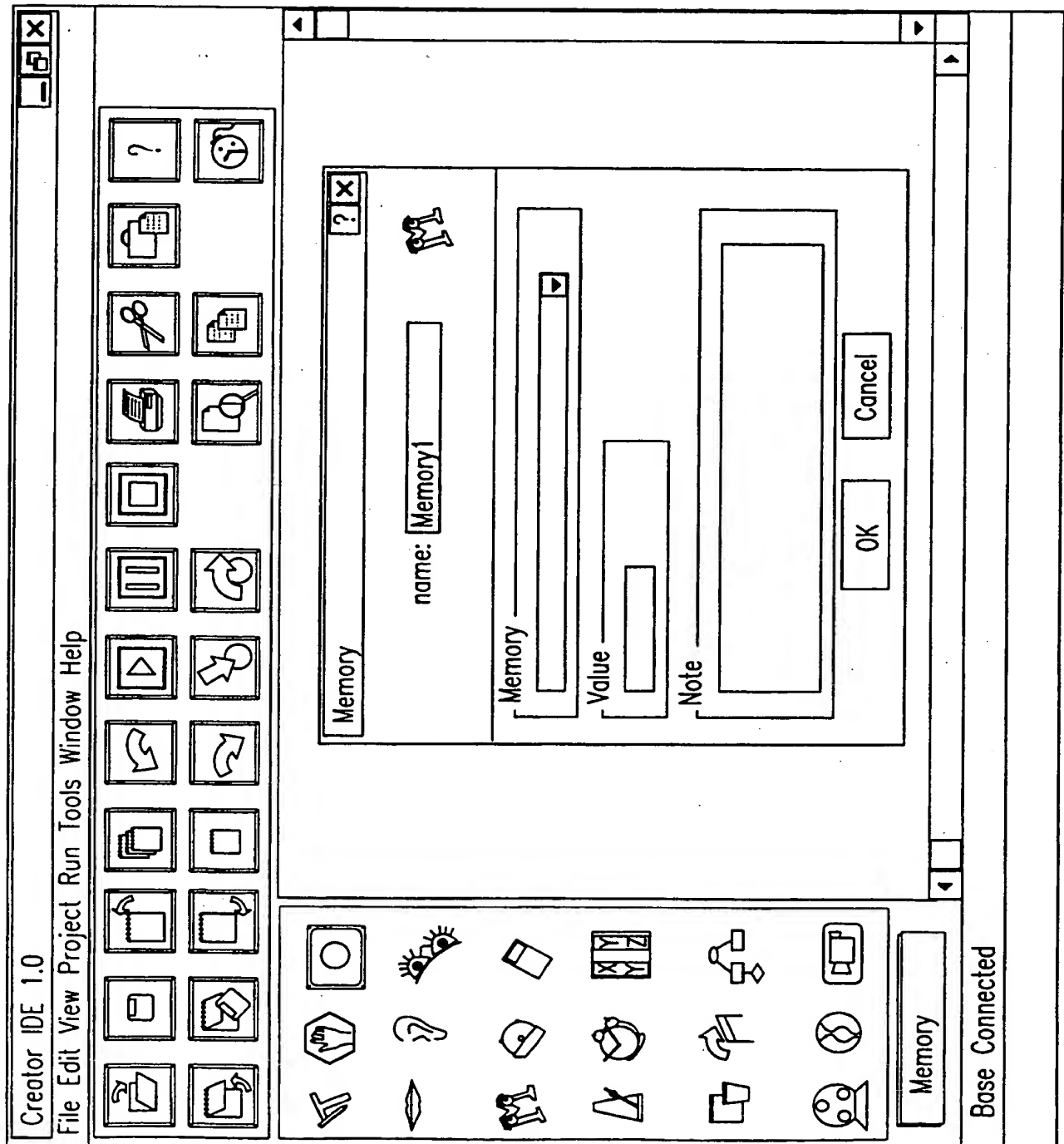
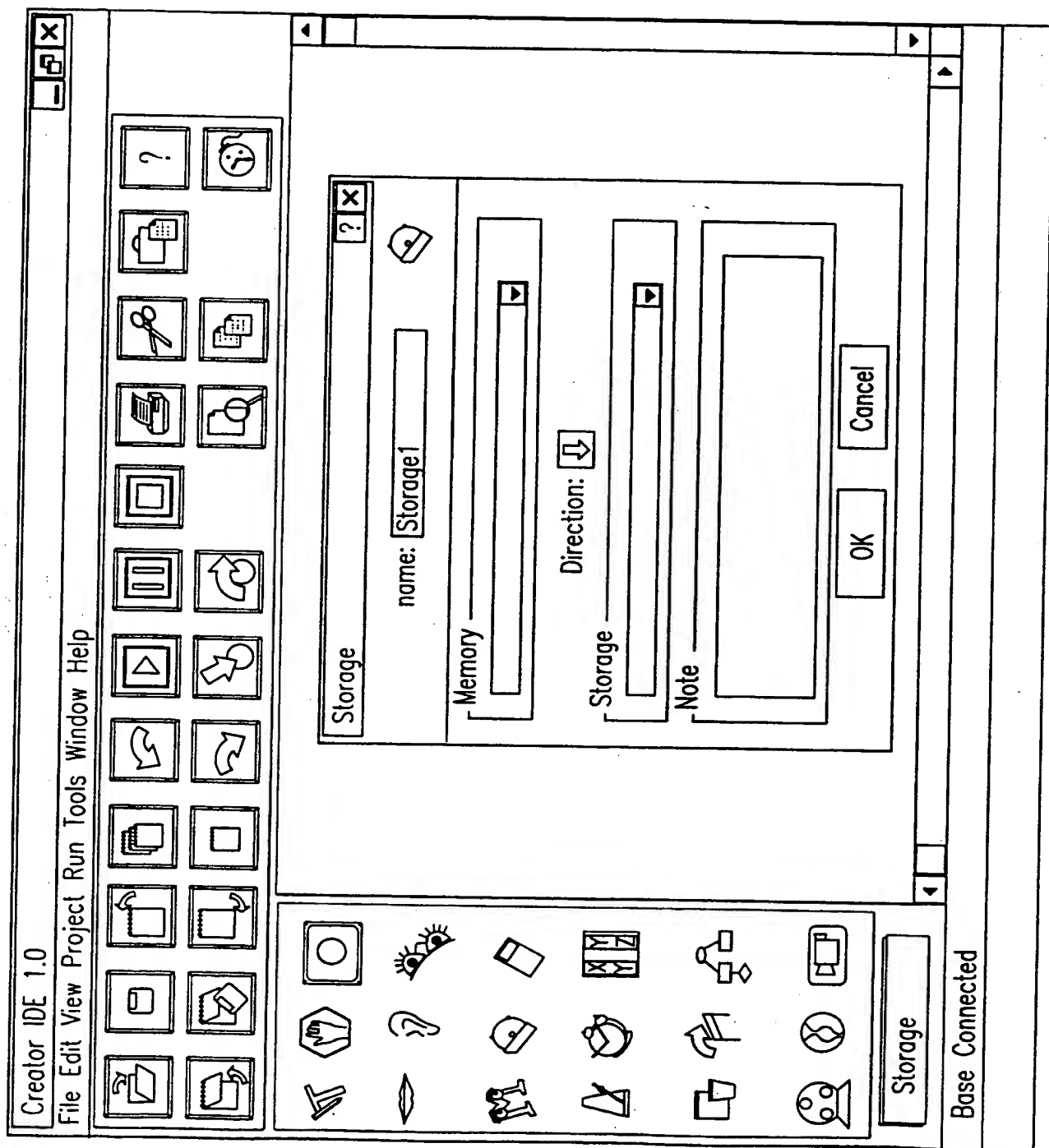
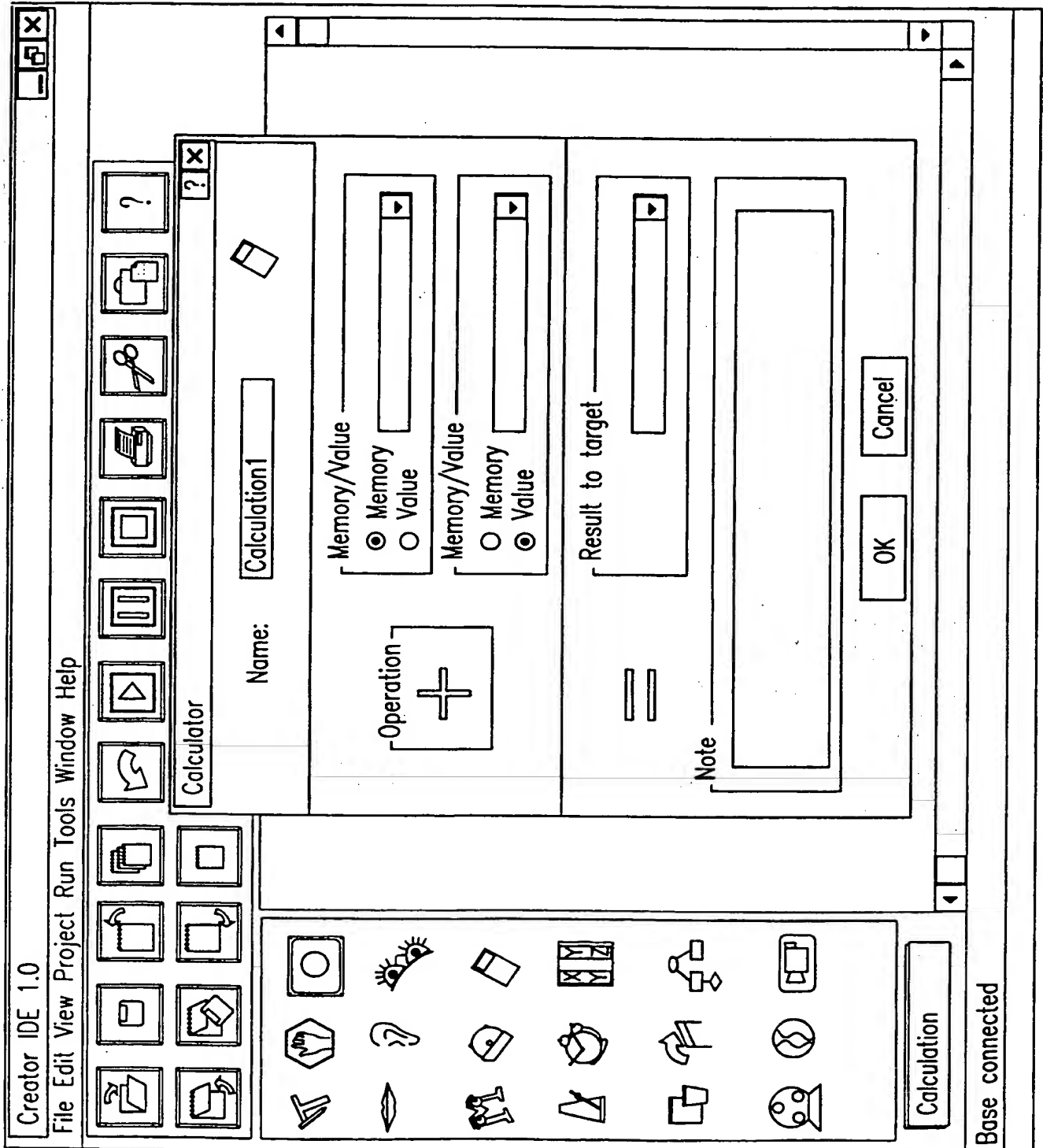


FIG. 16



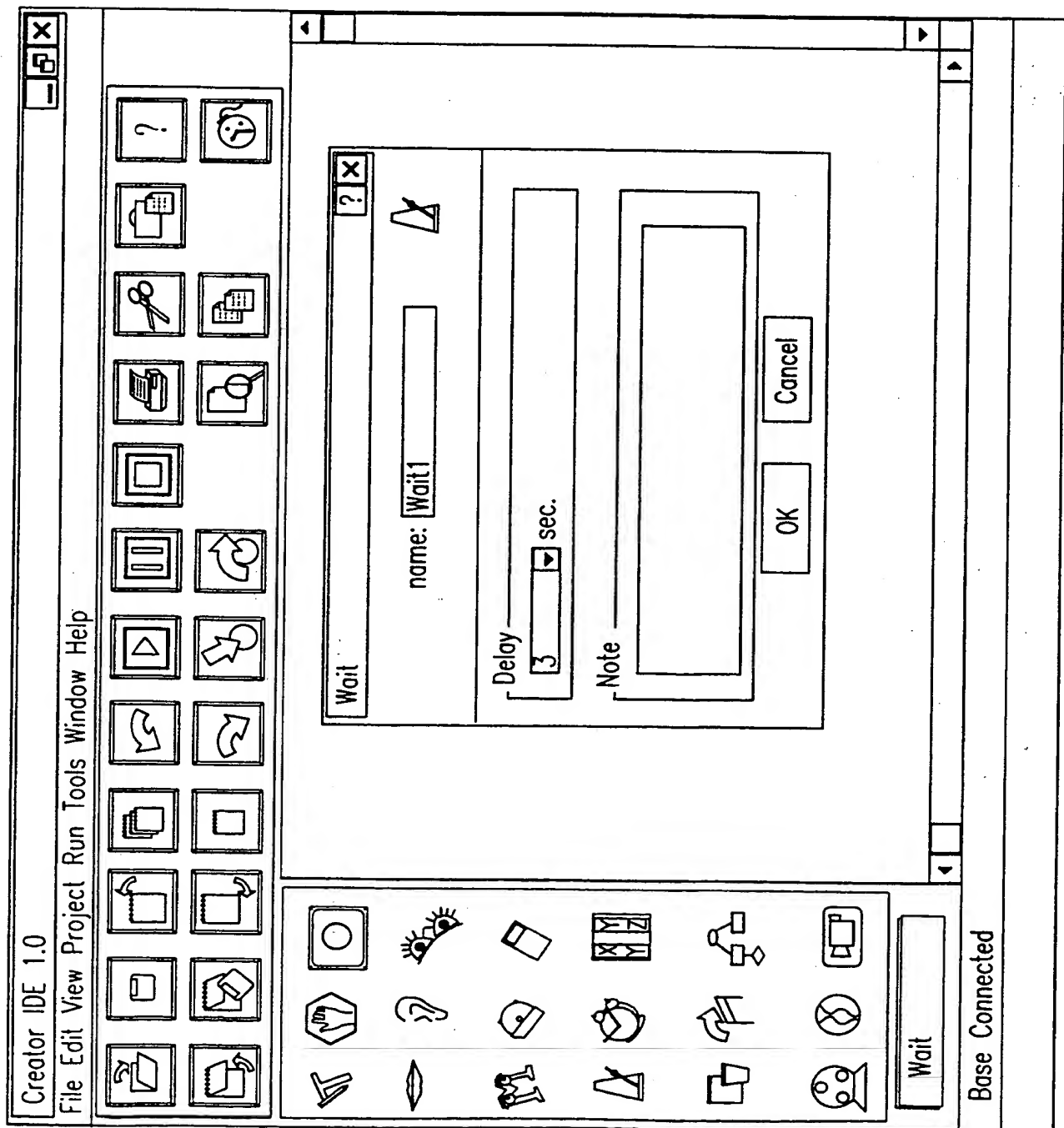
17/120

FIG. 17



18/120

FIG. 18



19/120

FIG. 19

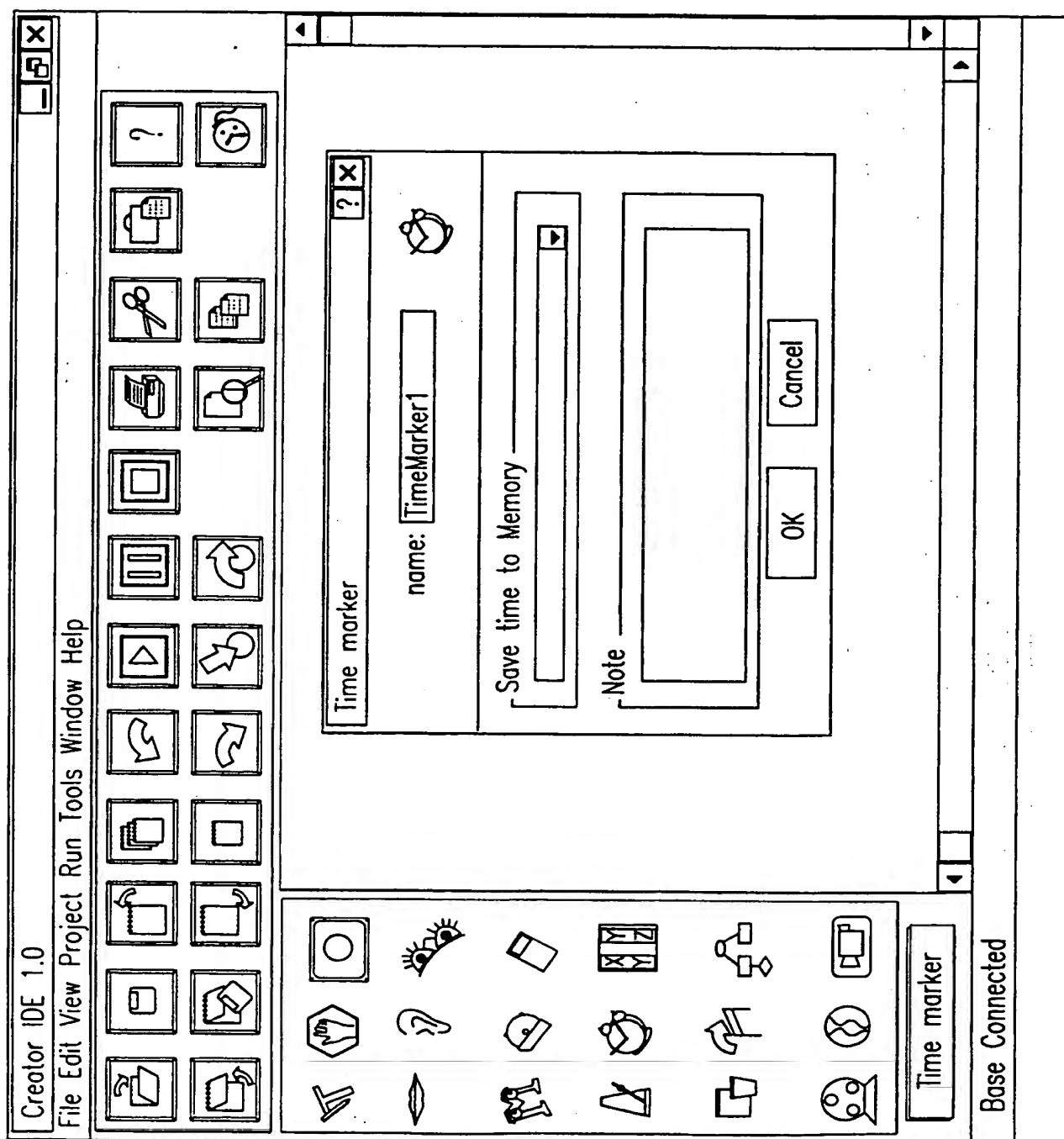
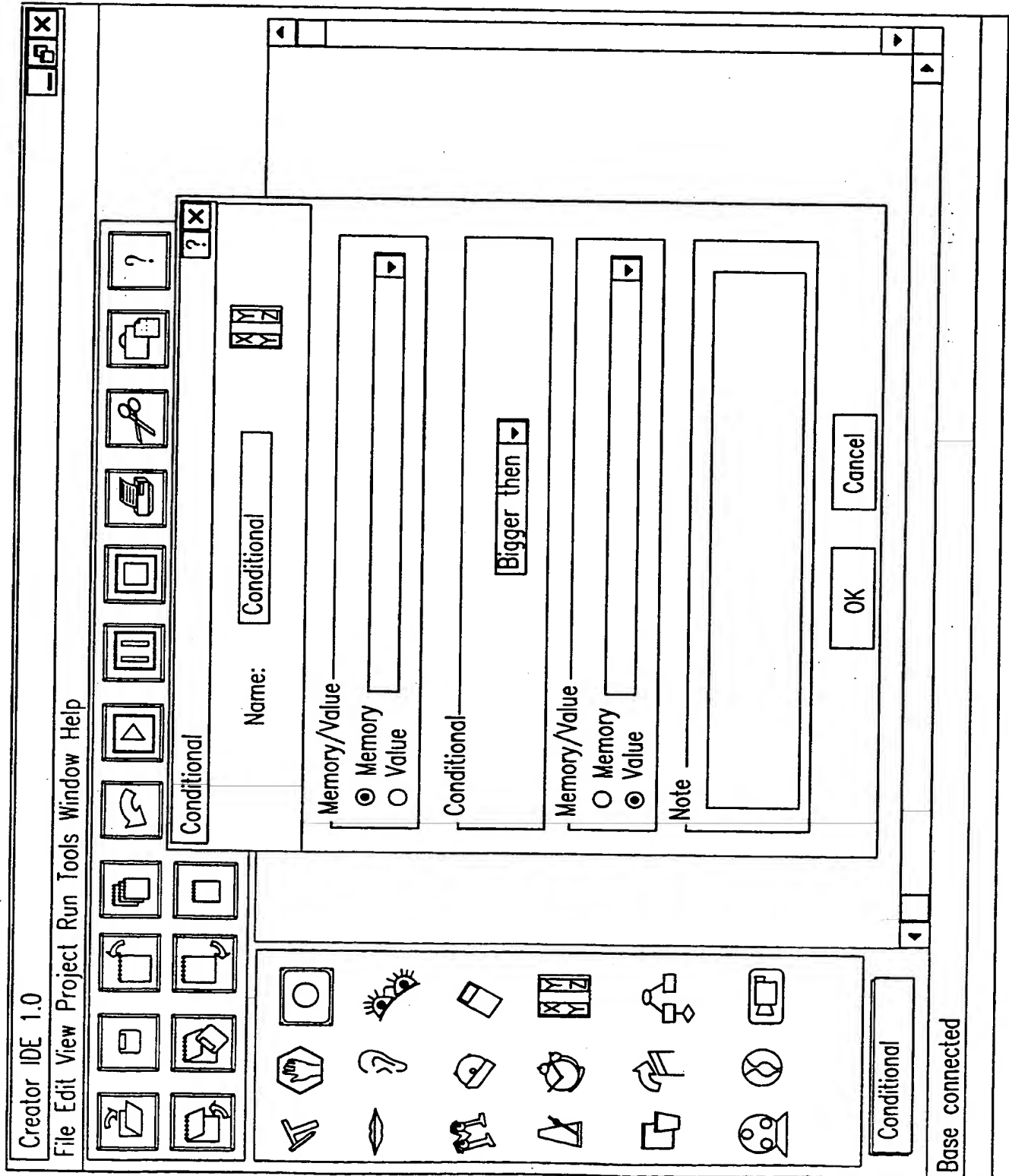


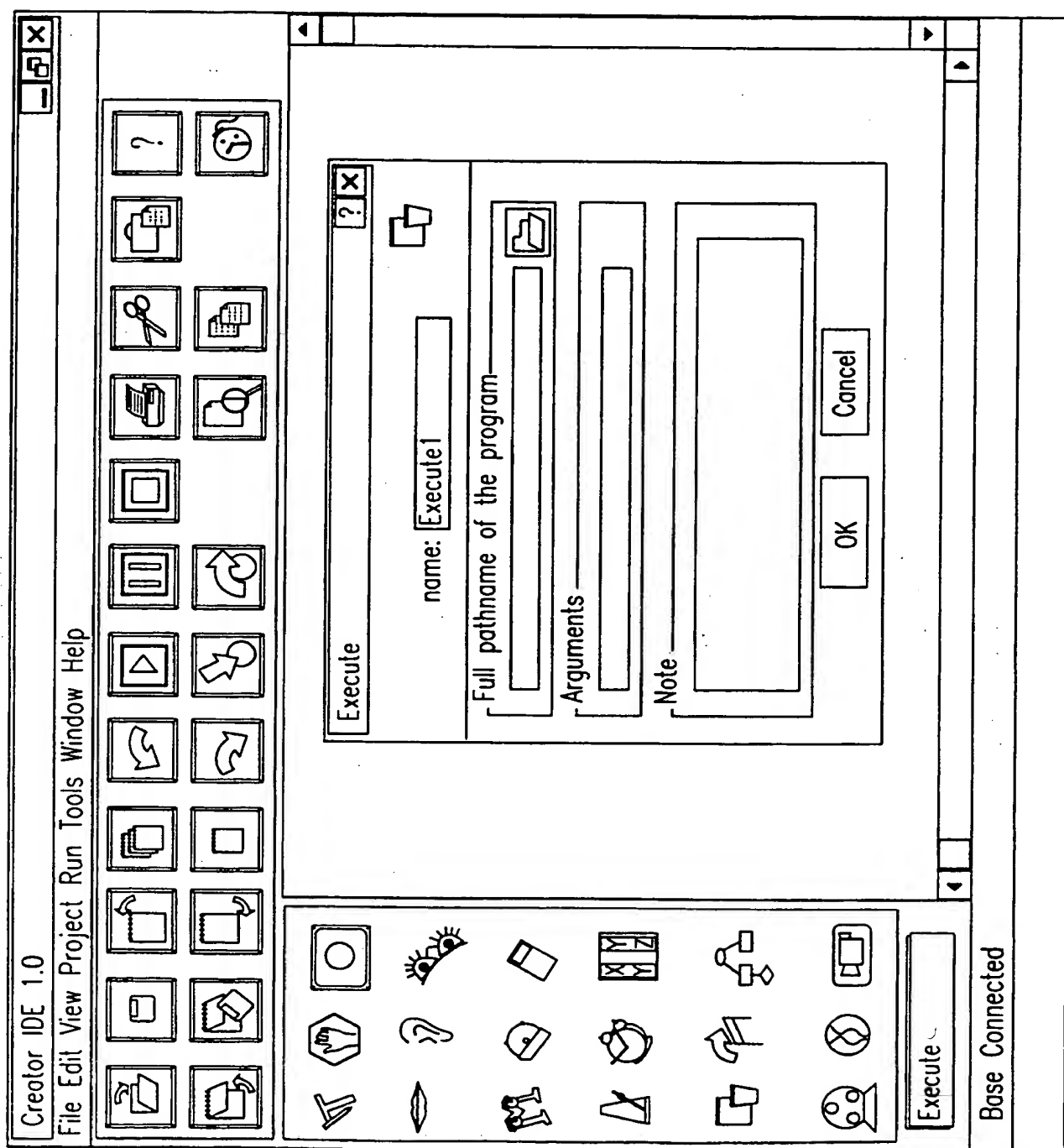
FIG. 20

20/120



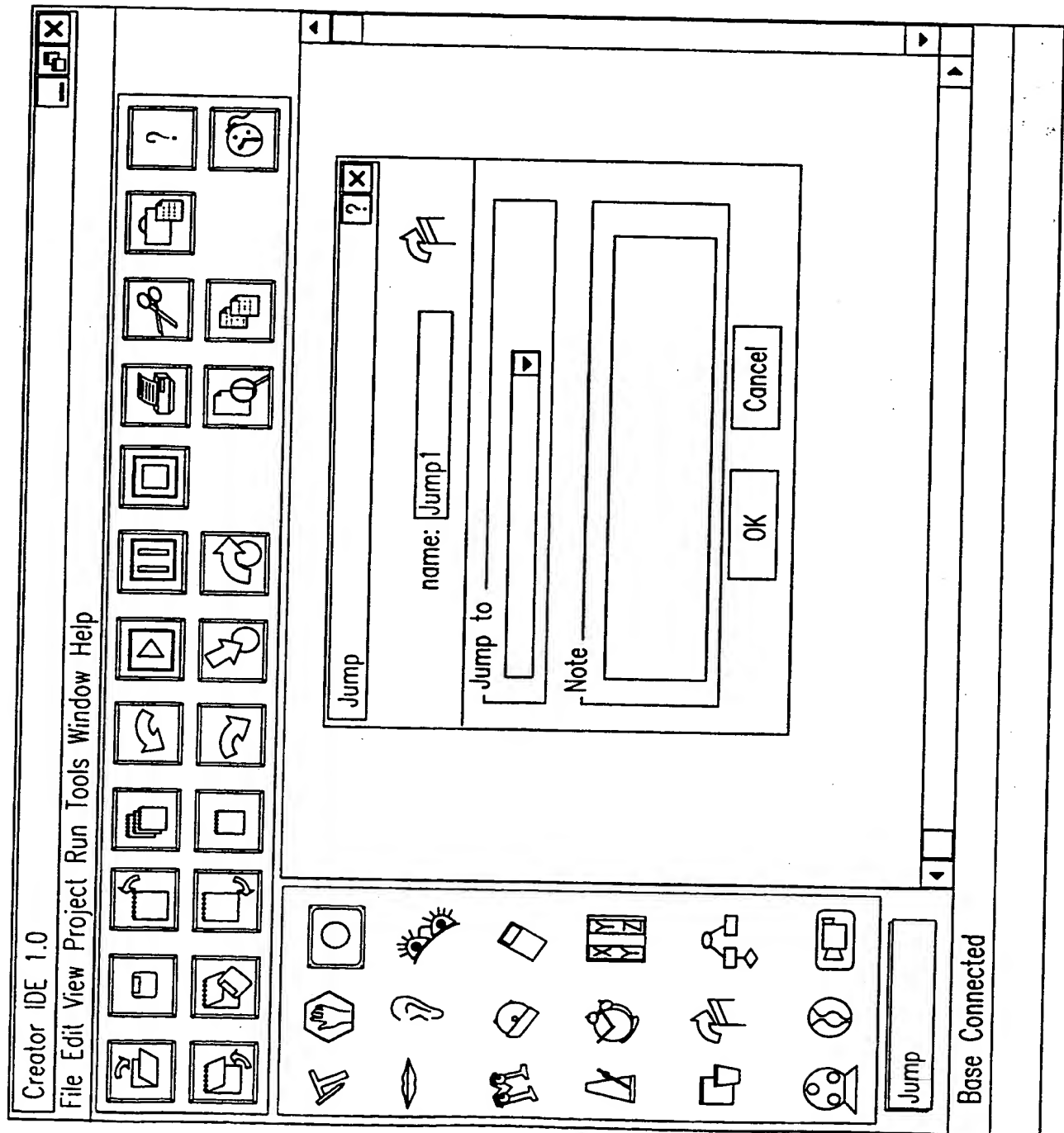
21/120

FIG. 21



22/120

FIG. 22



23/120

FIG. 23

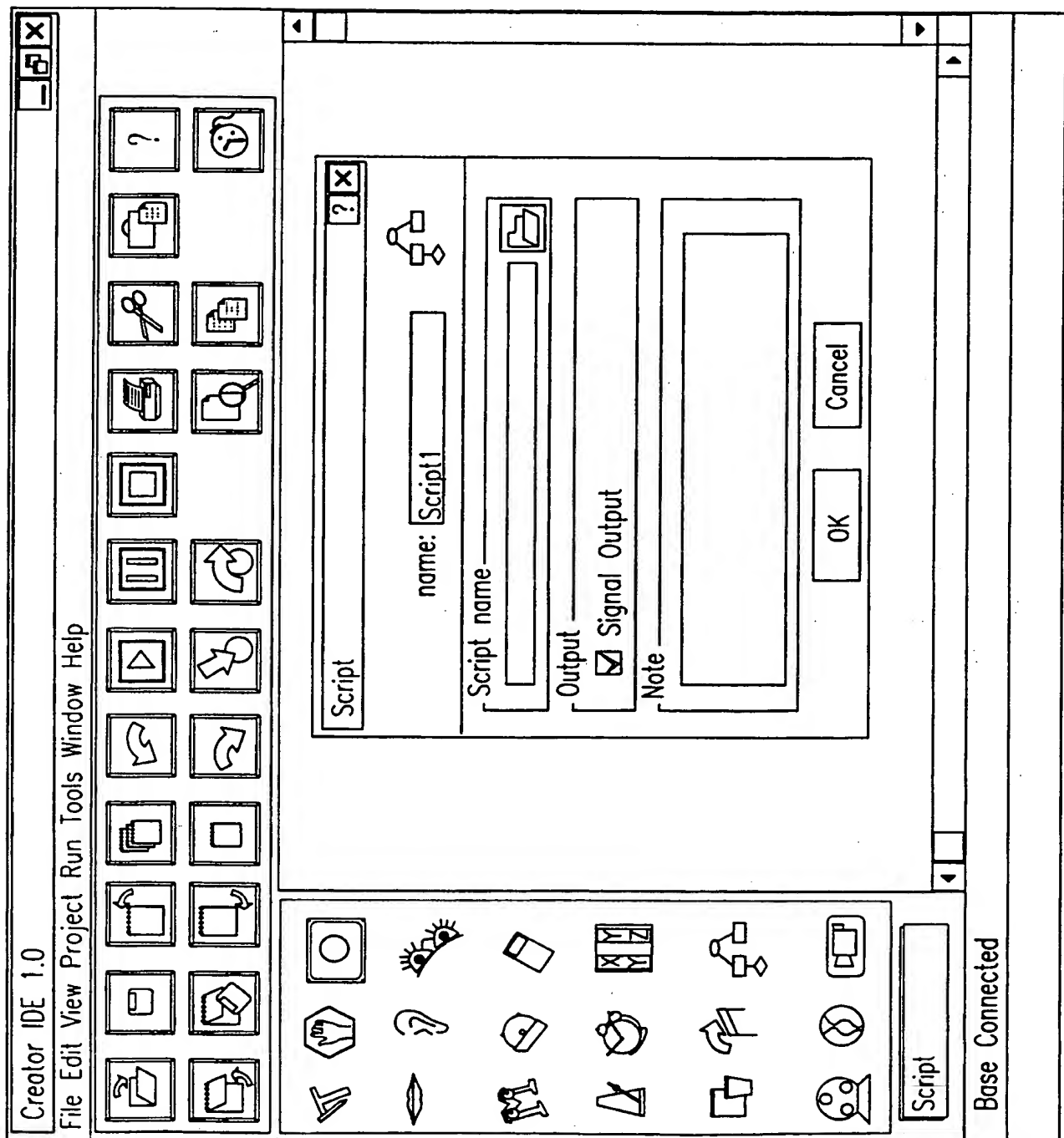
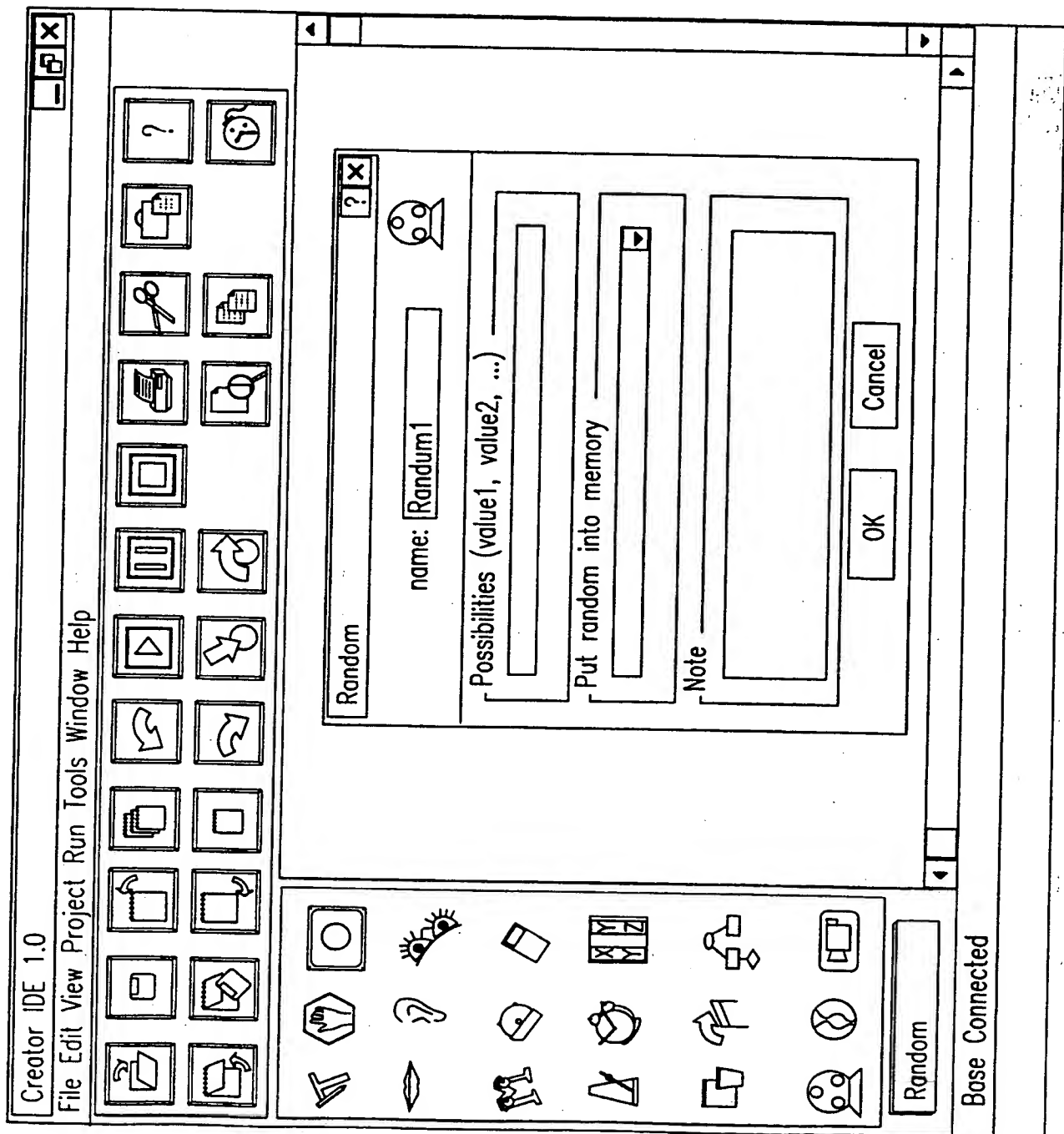
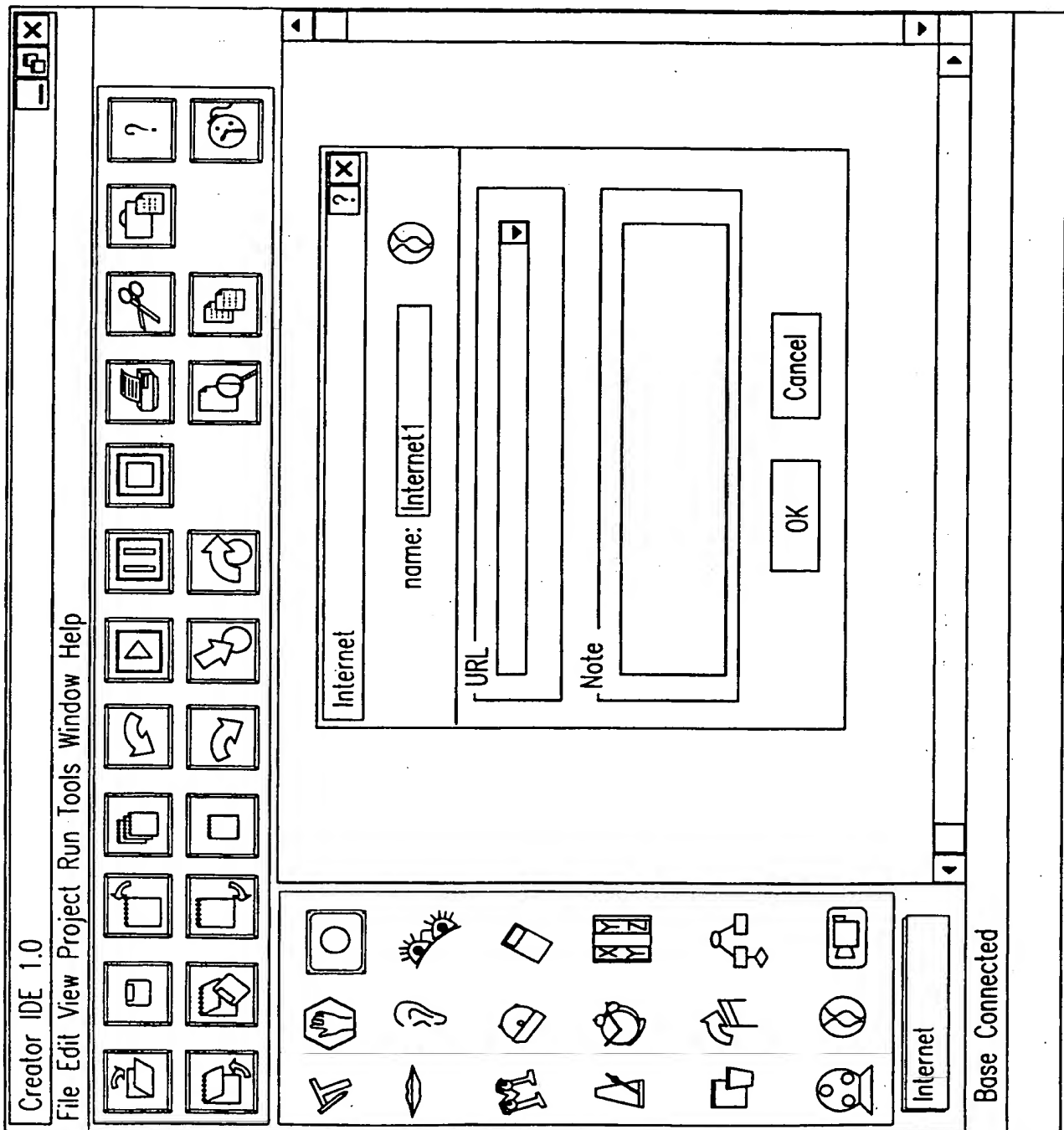


FIG. 24



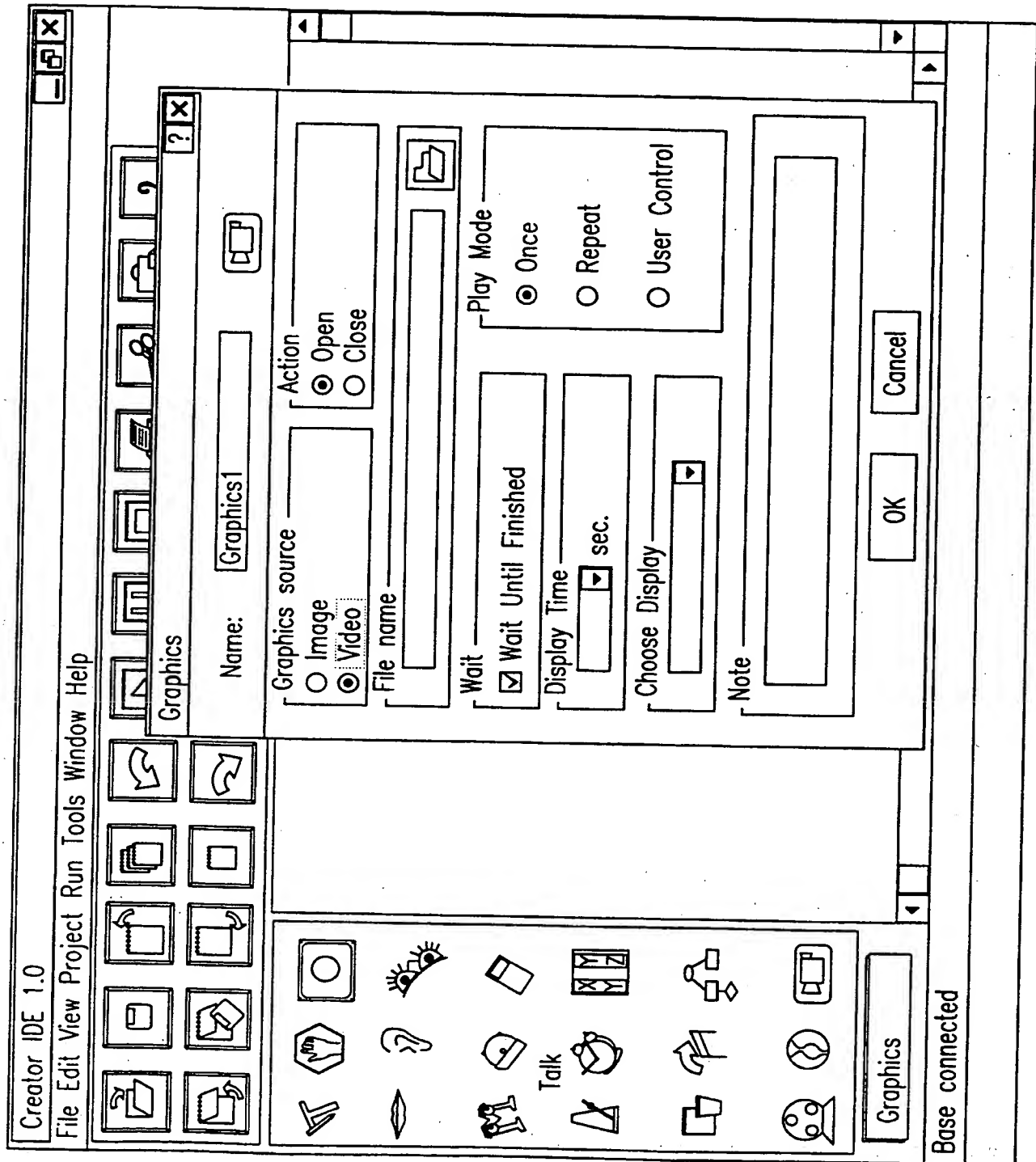
25/120

FIG. 25



26/120

FIG. 26



27/120

FIG. 27

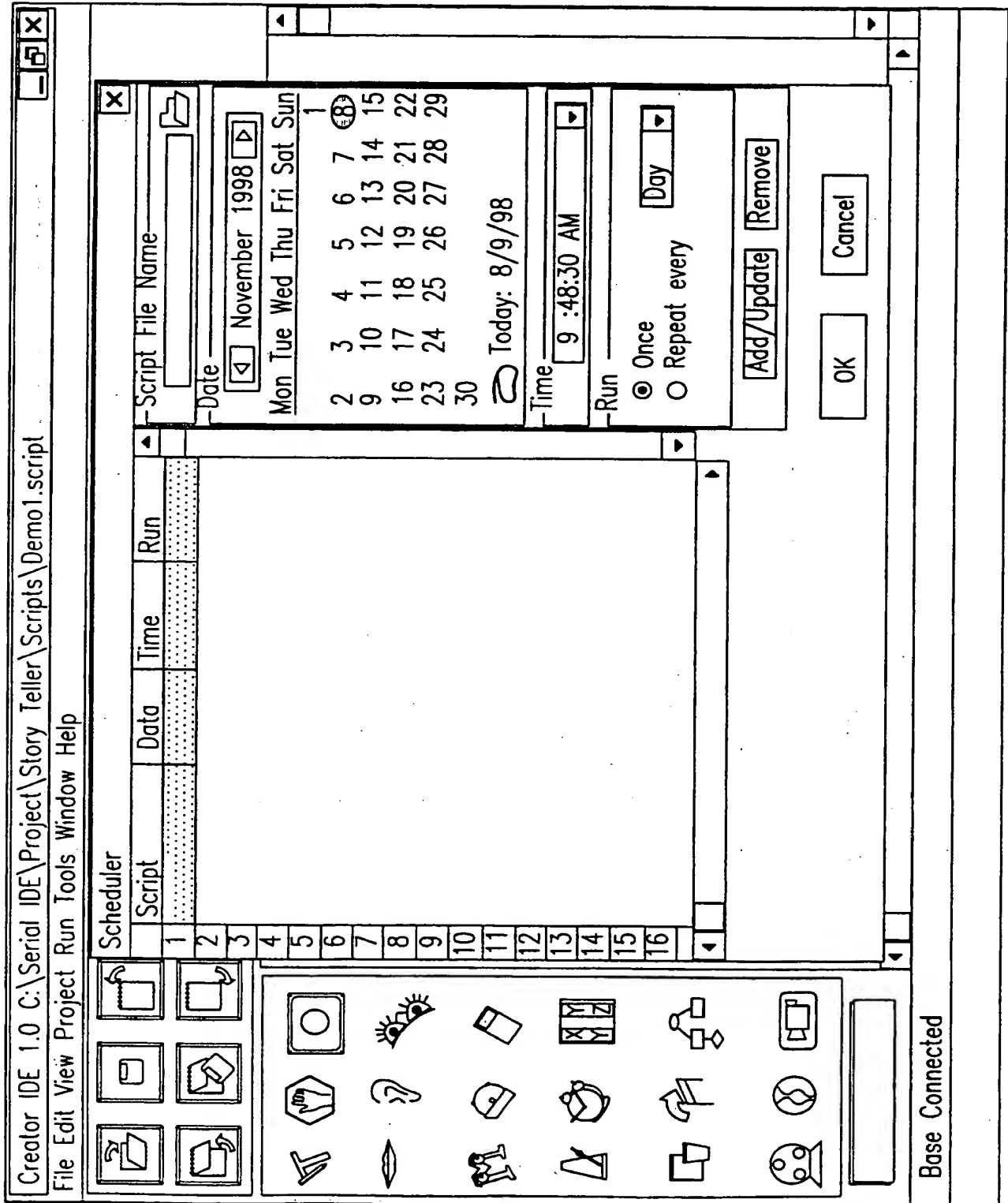


FIG. 28

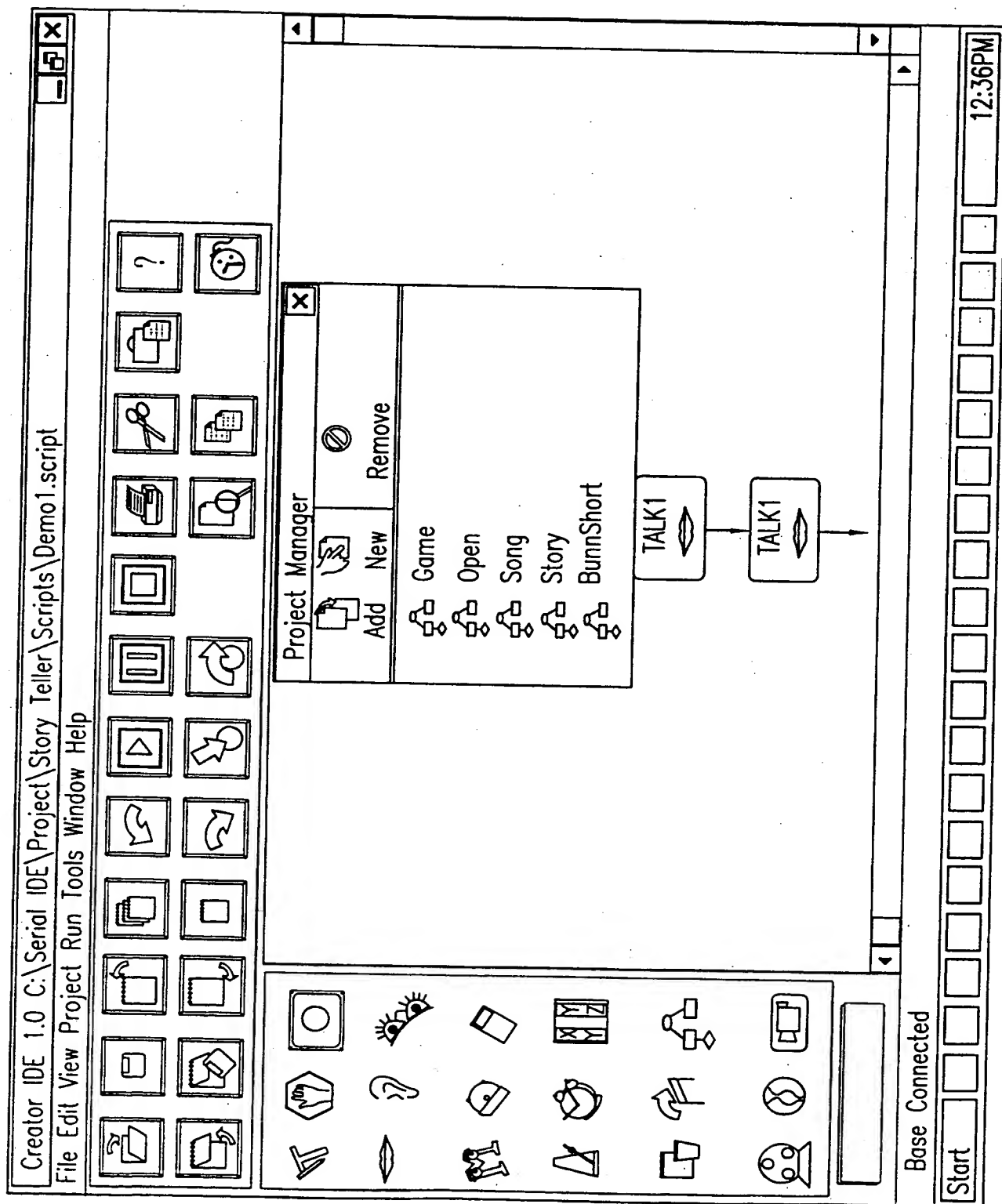


FIG. 29

29/120

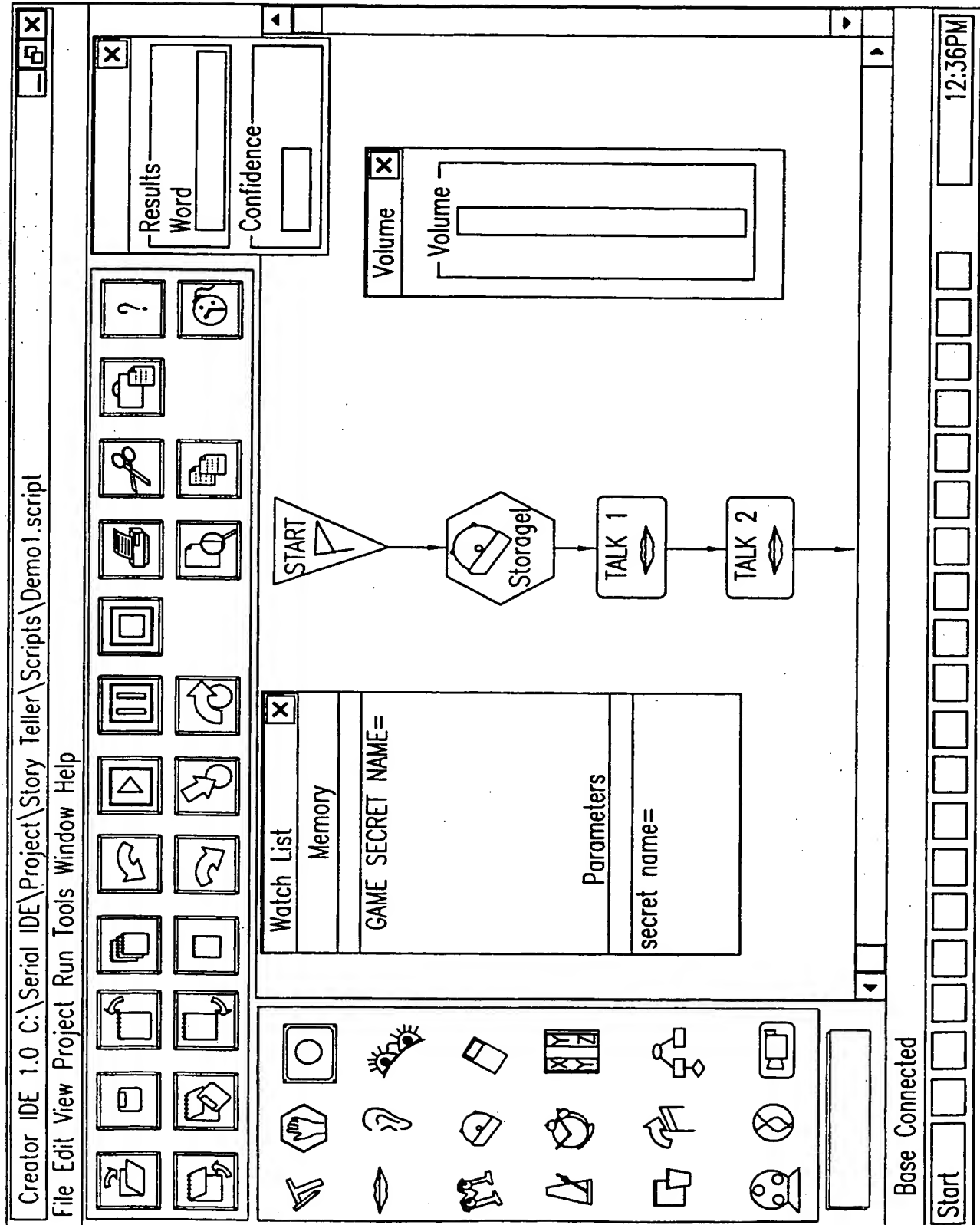
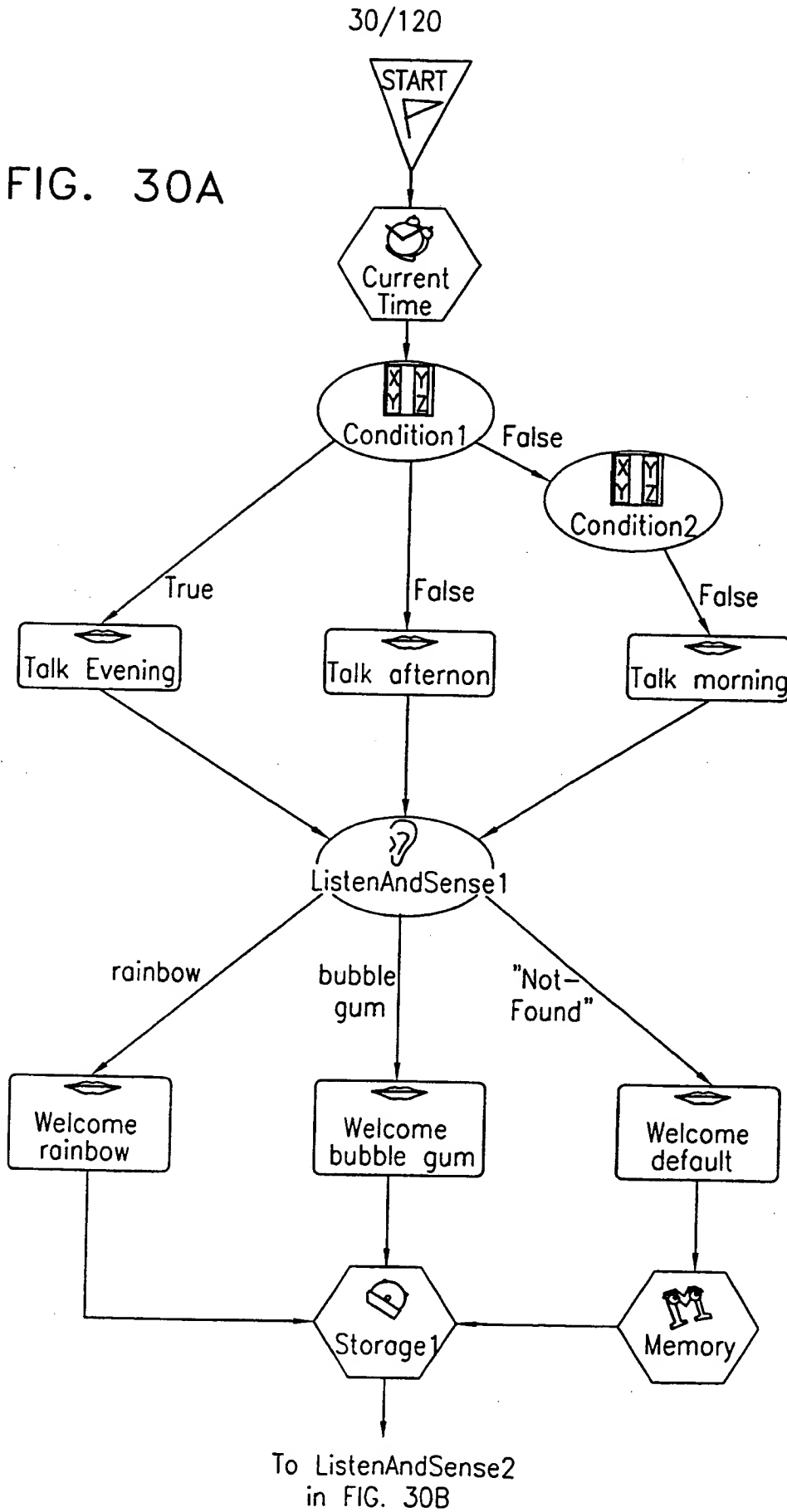
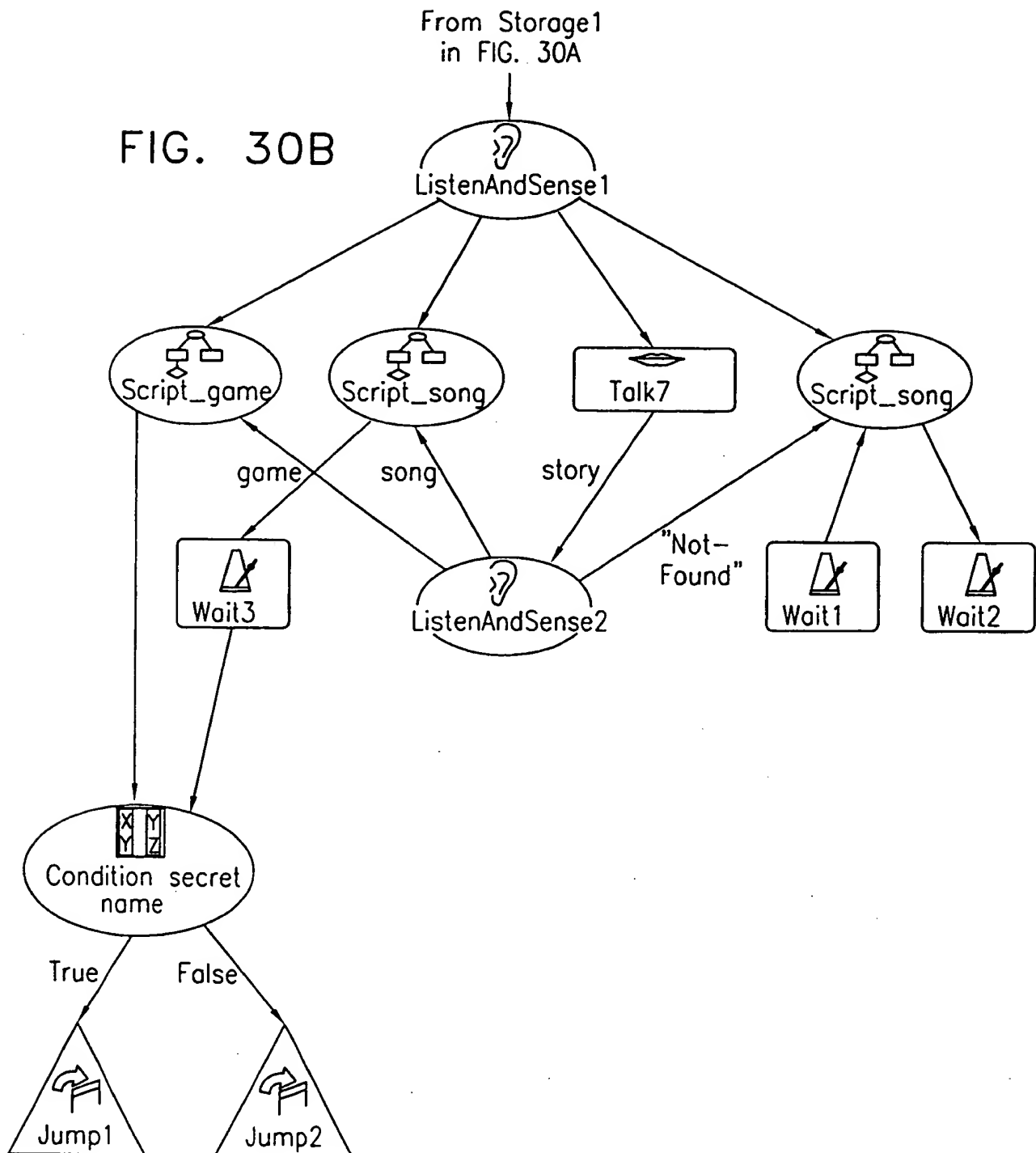


FIG. 30A



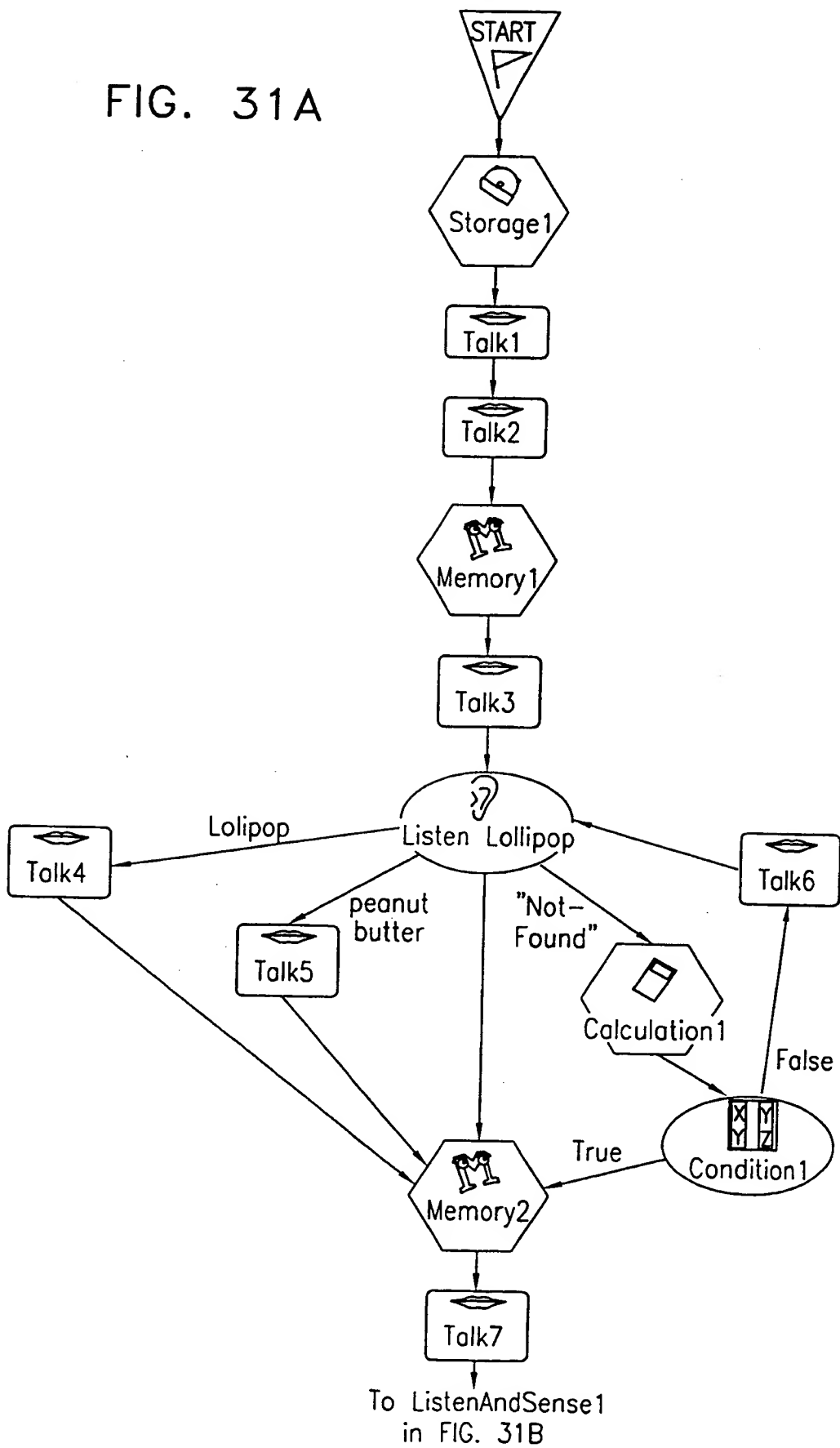
31/120

FIG. 30B



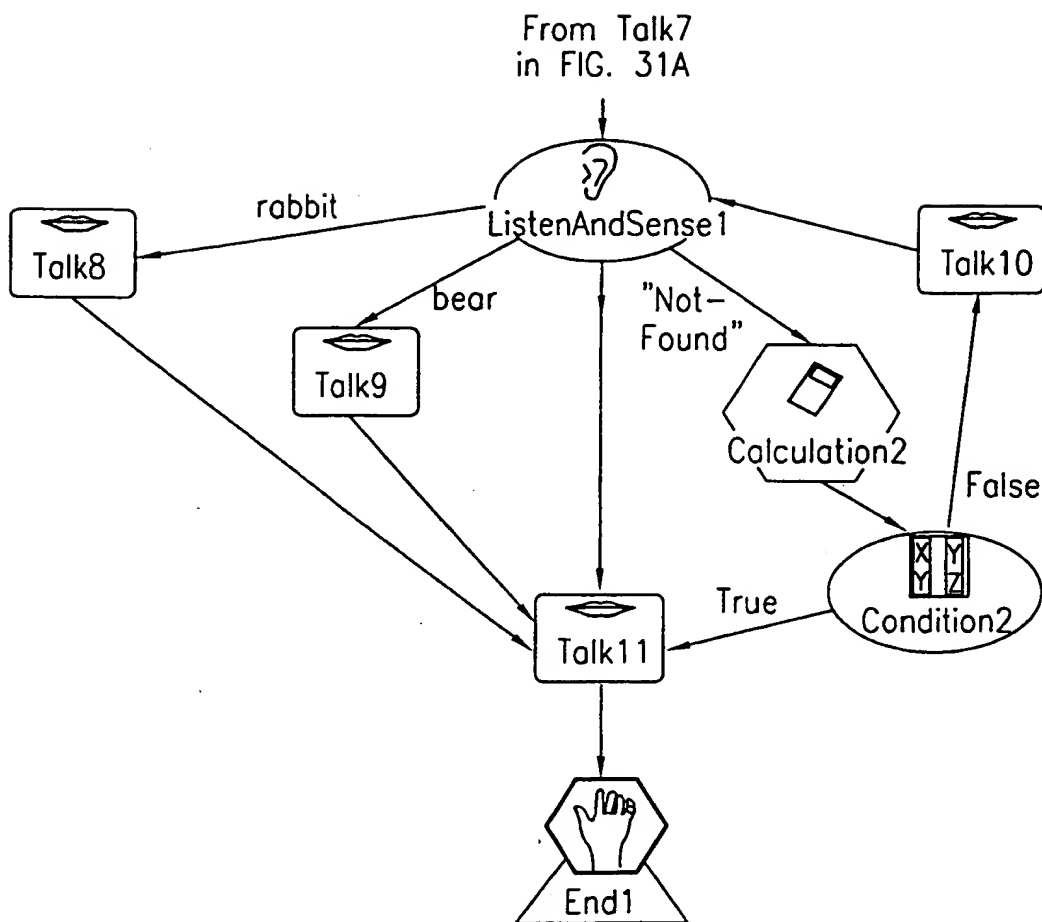
32/120

FIG. 31A



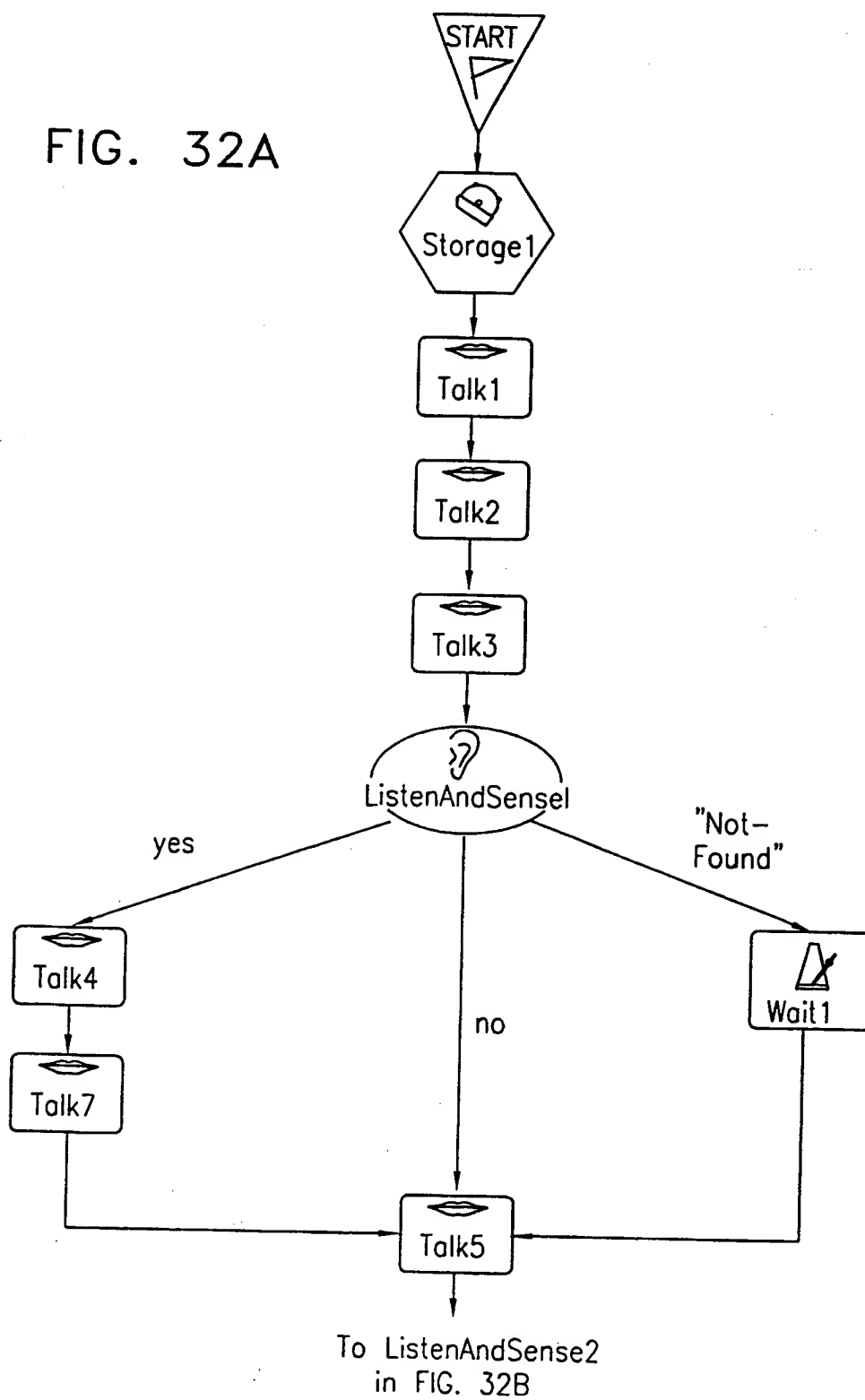
33/120

FIG. 31B



34/120

FIG. 32A



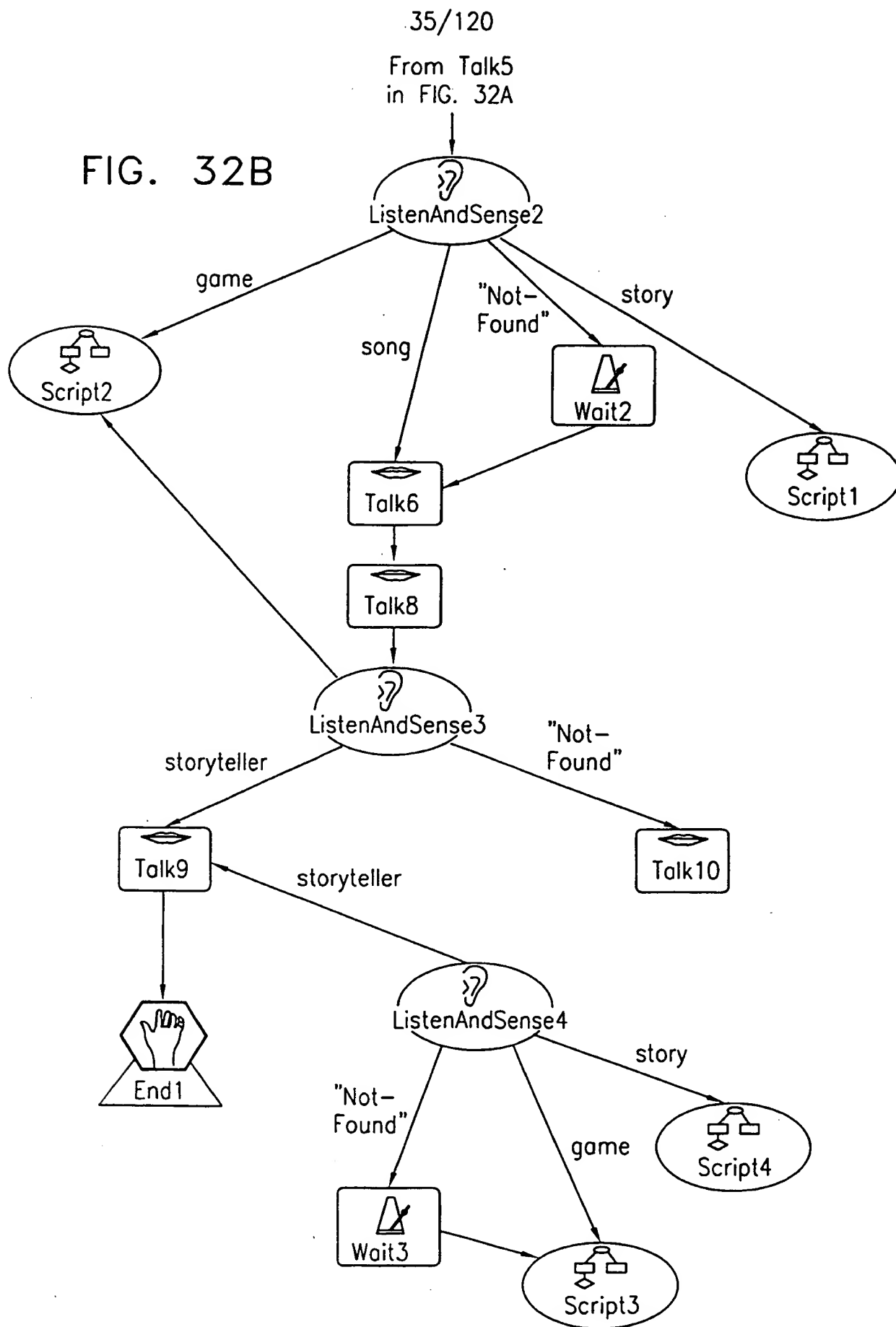
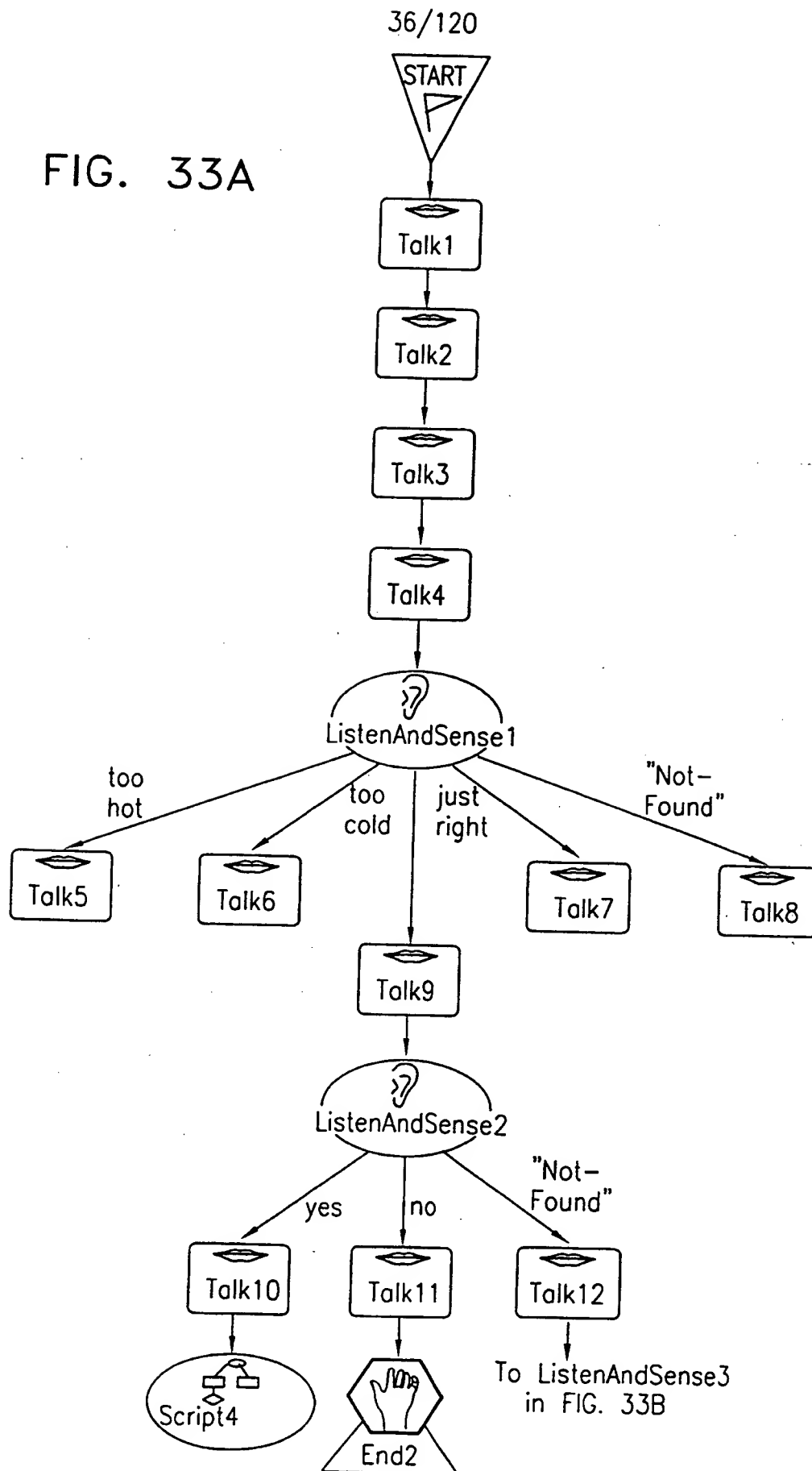
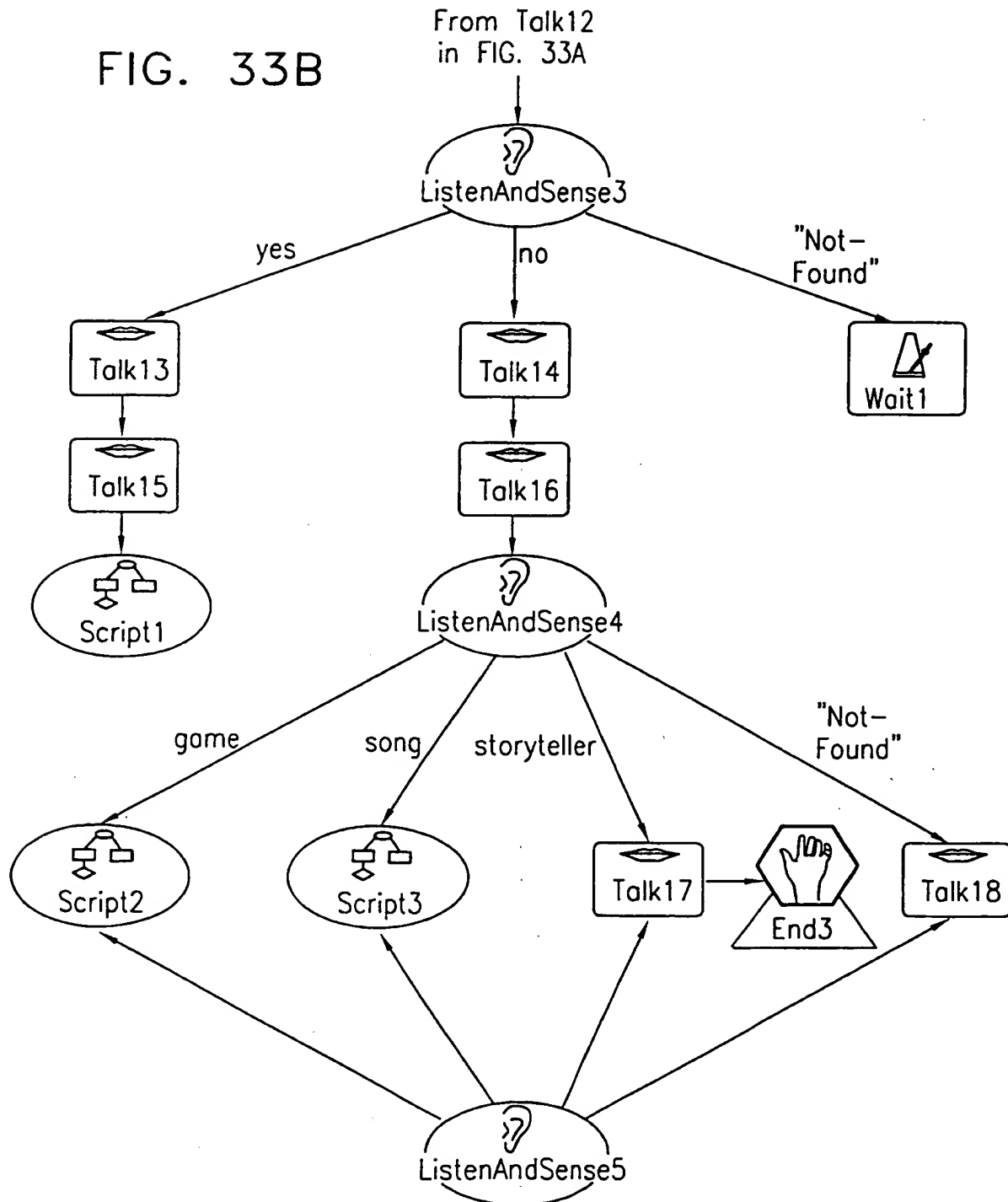


FIG. 33A



37/120

FIG. 33B



38/120

FIG. 34

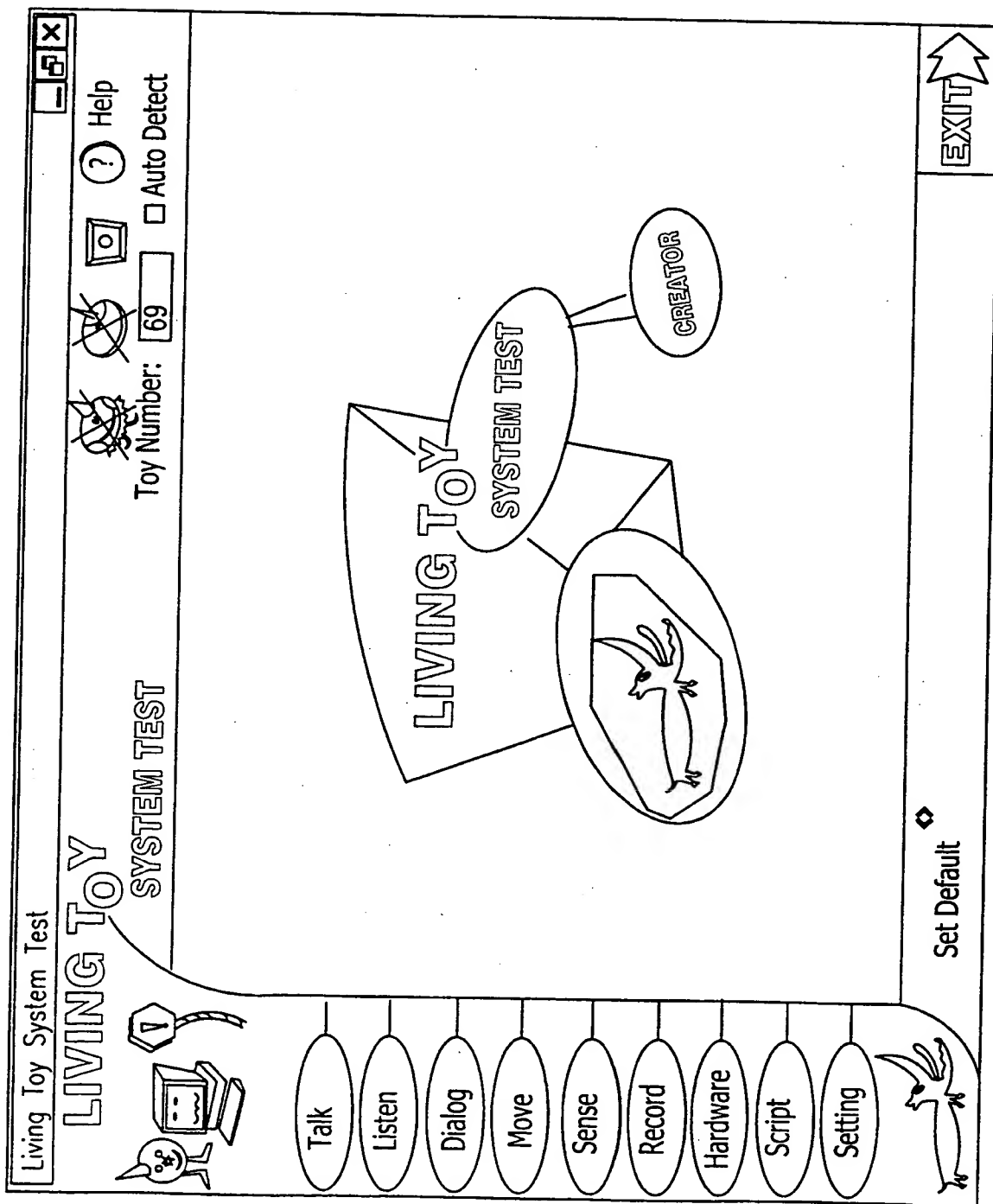


FIG. 35

LIVING TOY

Living Toy System Test

SYSTEM TEST

Help ? Auto Detect ☐

Toy Number: 70

Configure

WAV file:

Text-To-Speech:

Continuous operation

Type of movement:

Talking time: sec.

Set Default

Talk Listen Dialog Move Sense Record Hardware Script Setting

40/120

FIG. 36

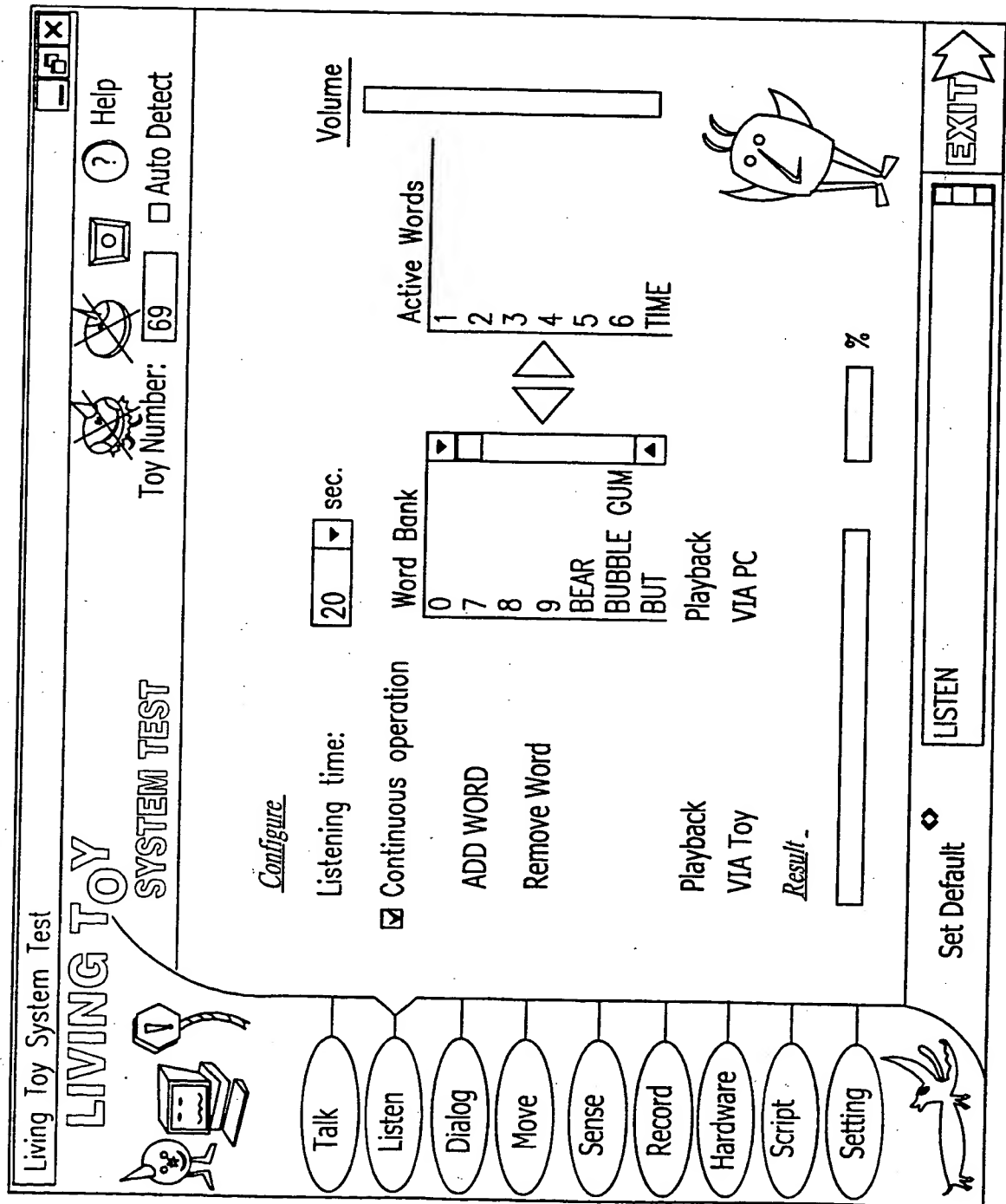
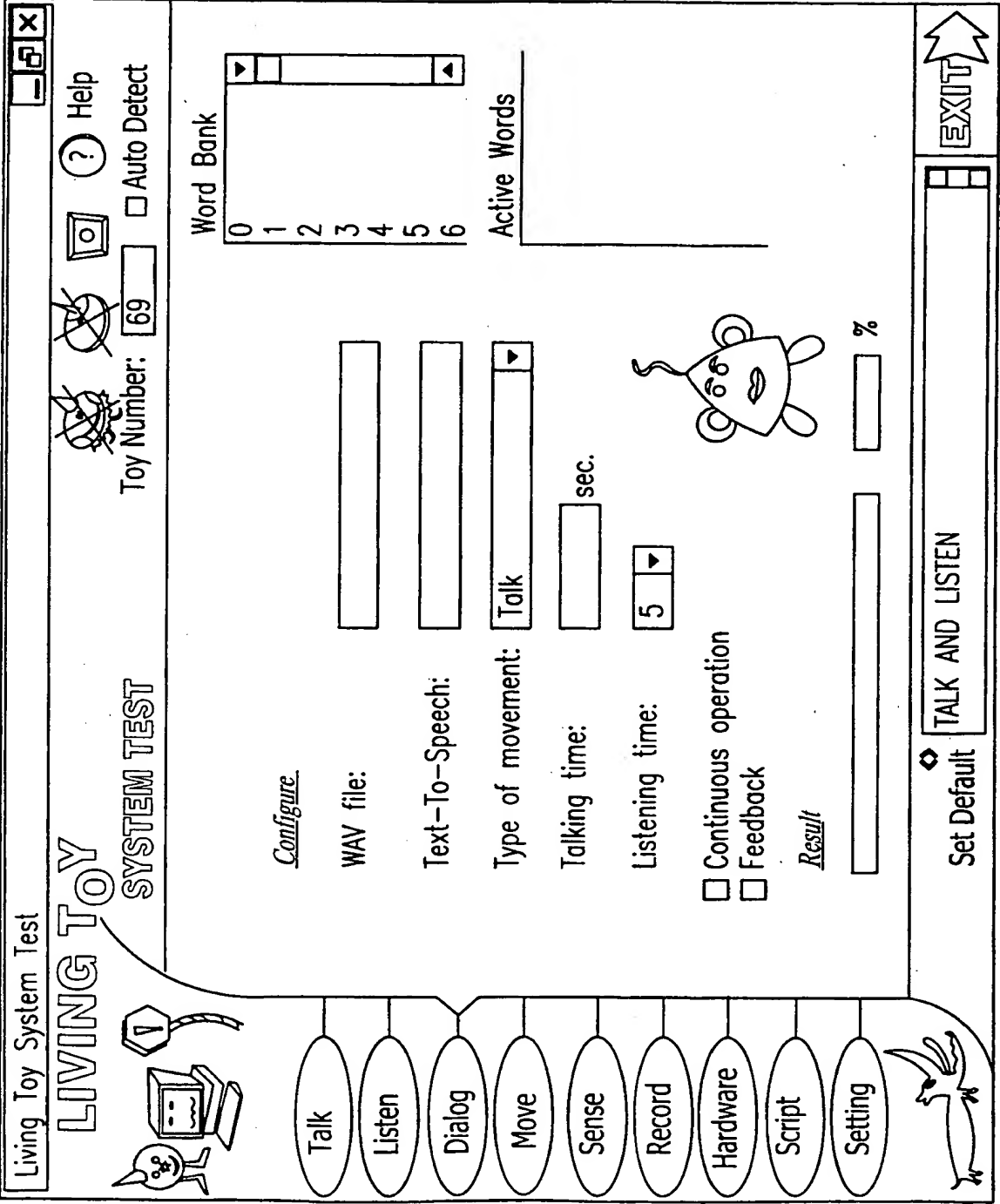
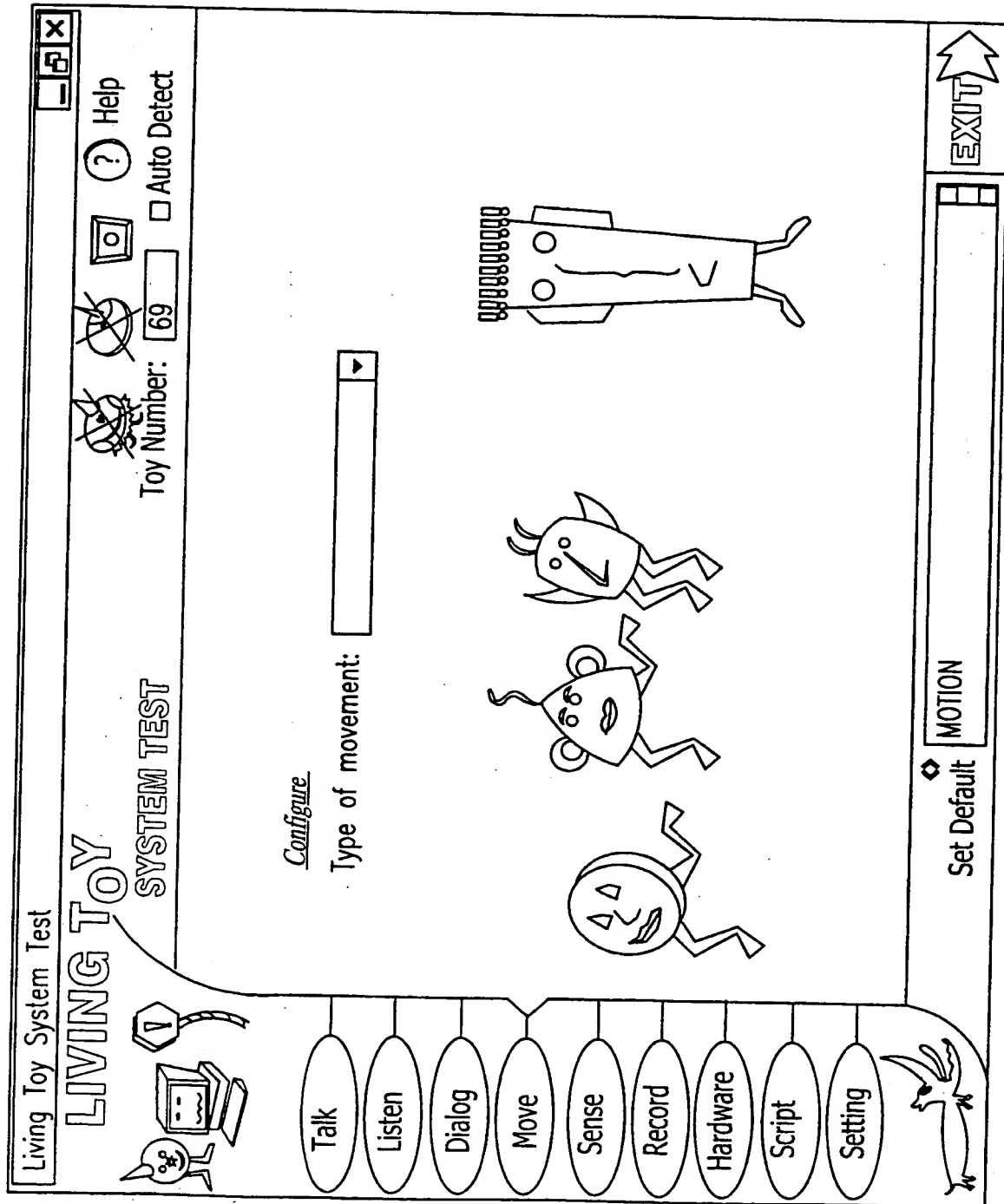


FIG. 37



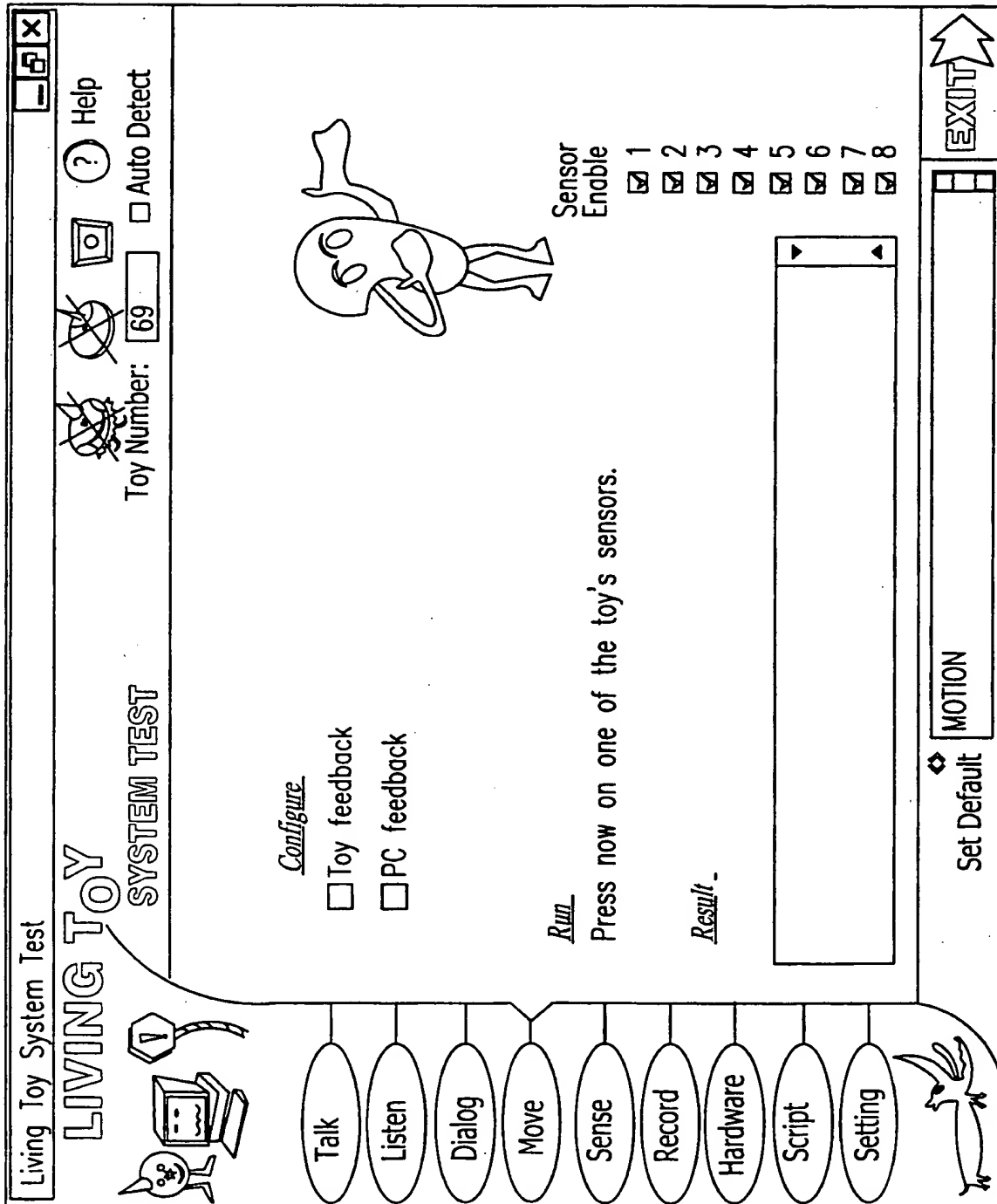
42/120

FIG. 38



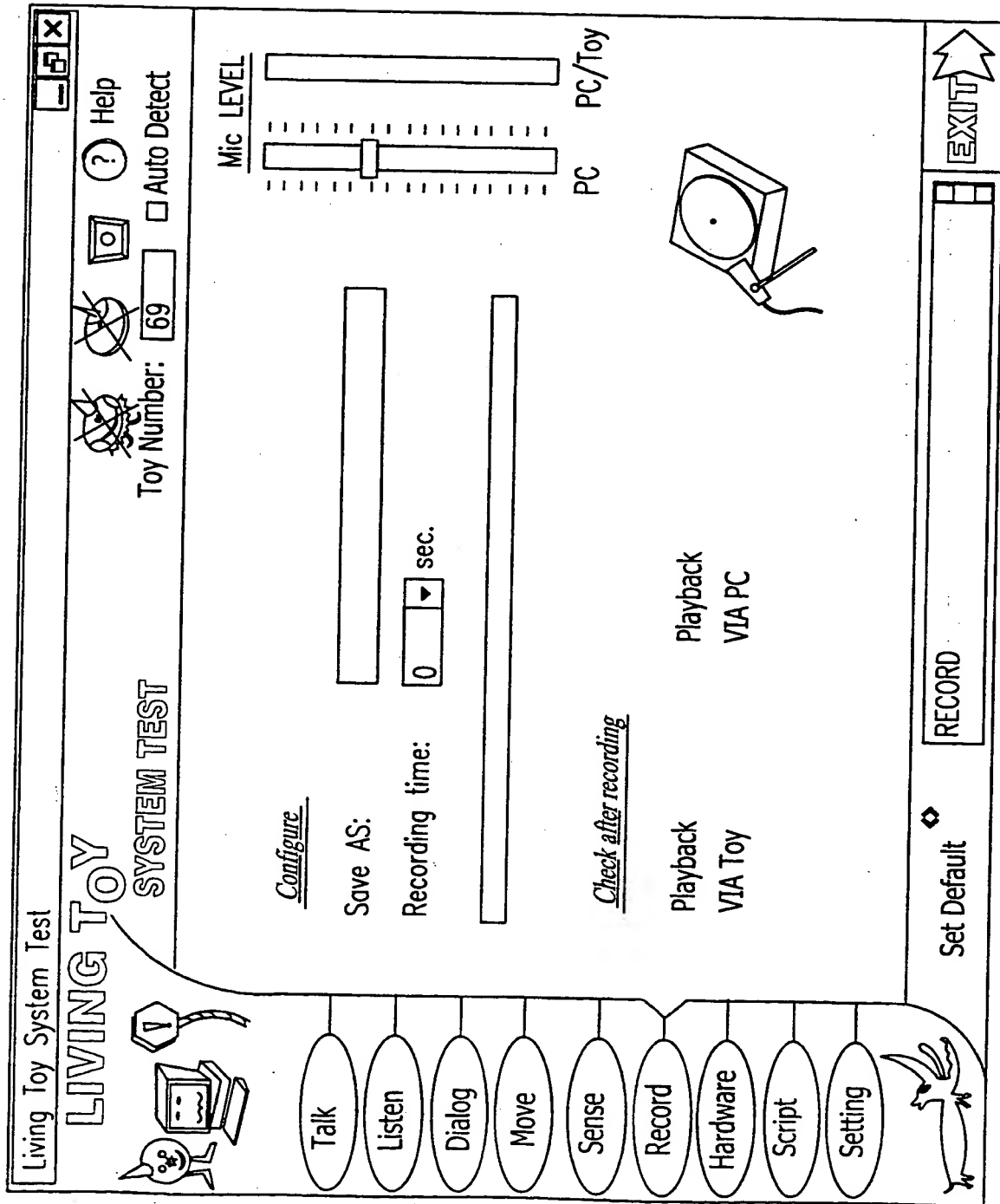
43/120

FIG. 39



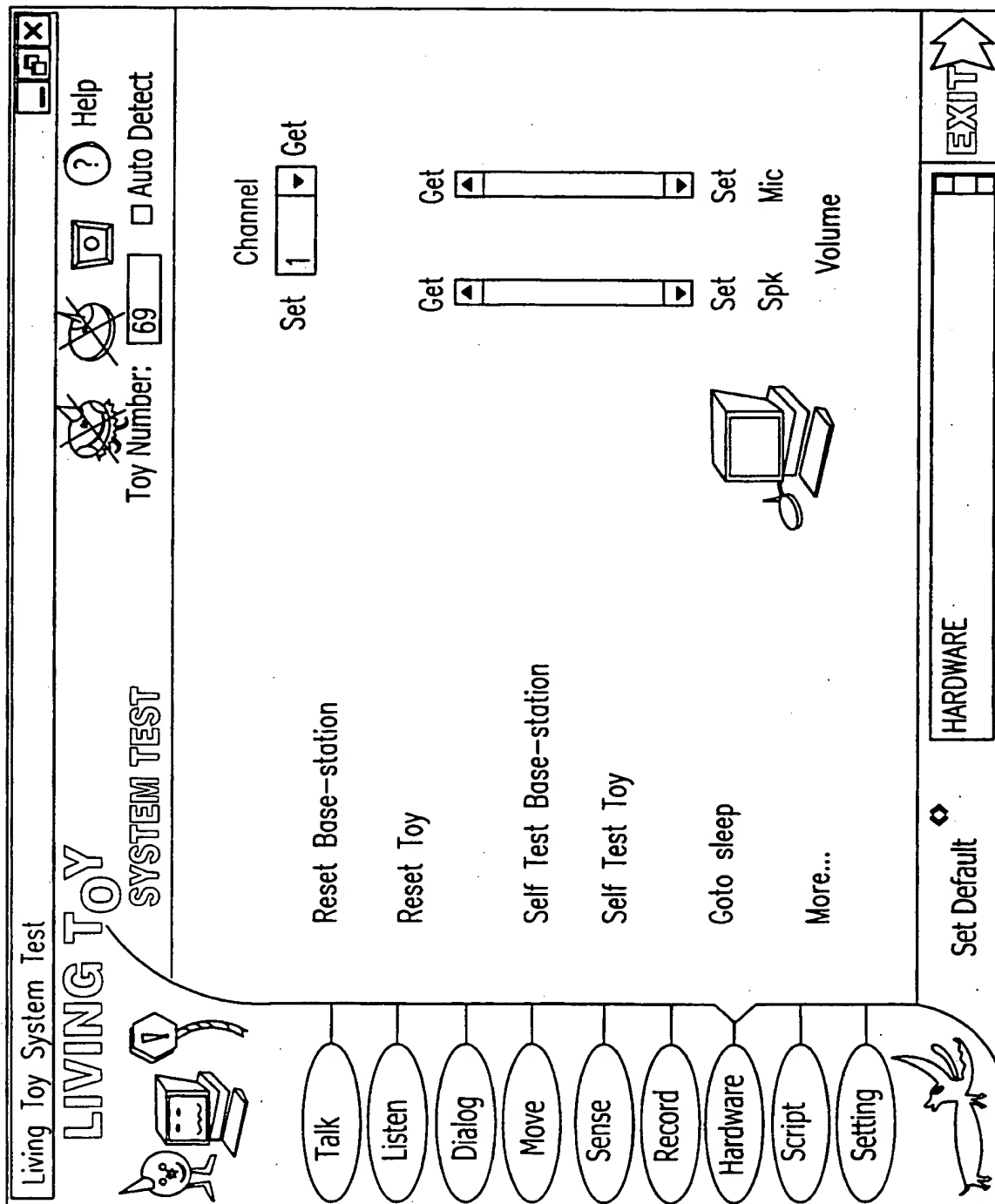
44/120

FIG. 40



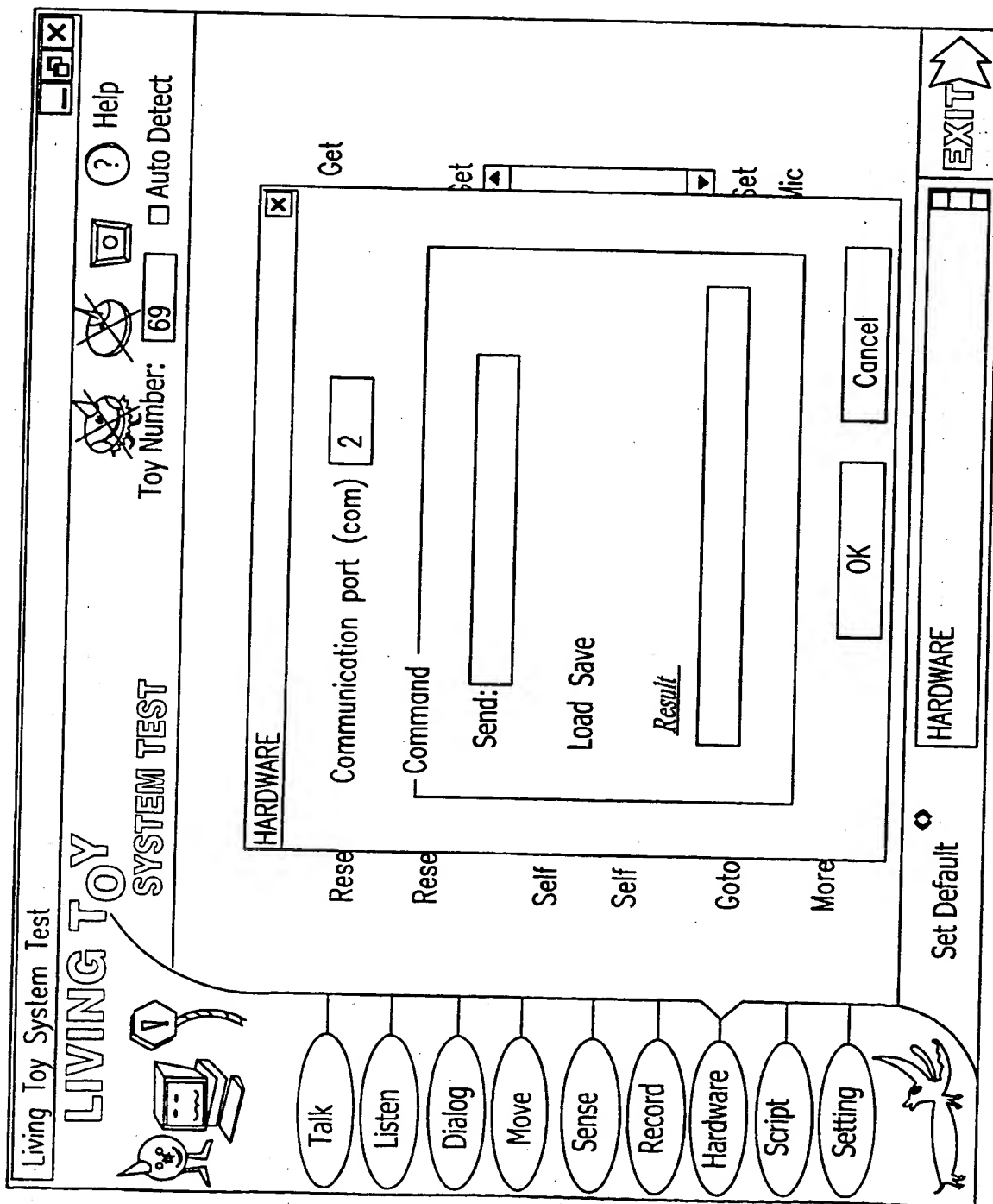
45/120

FIG. 41



46/120

FIG. 42



47/120

FIG. 43

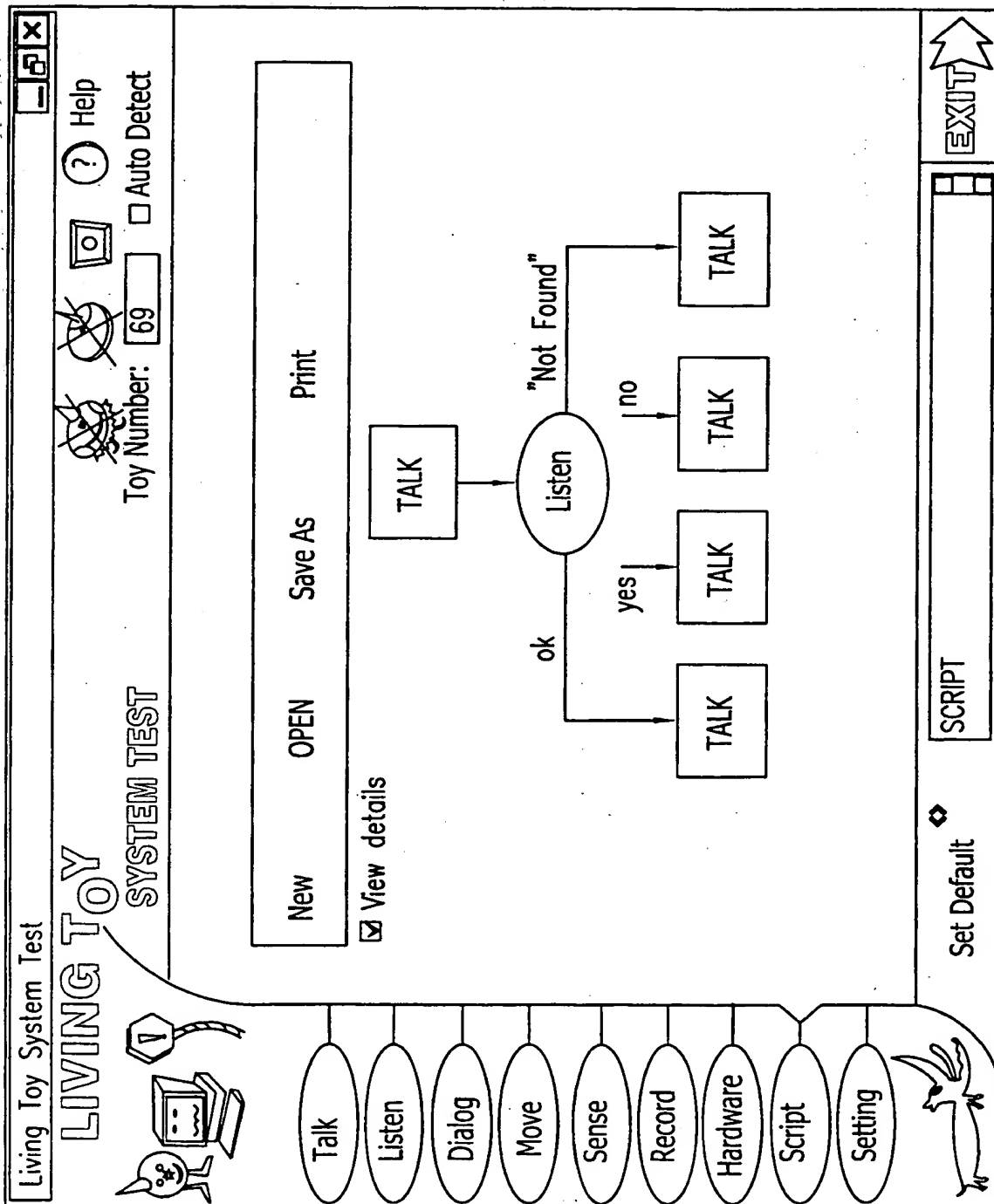
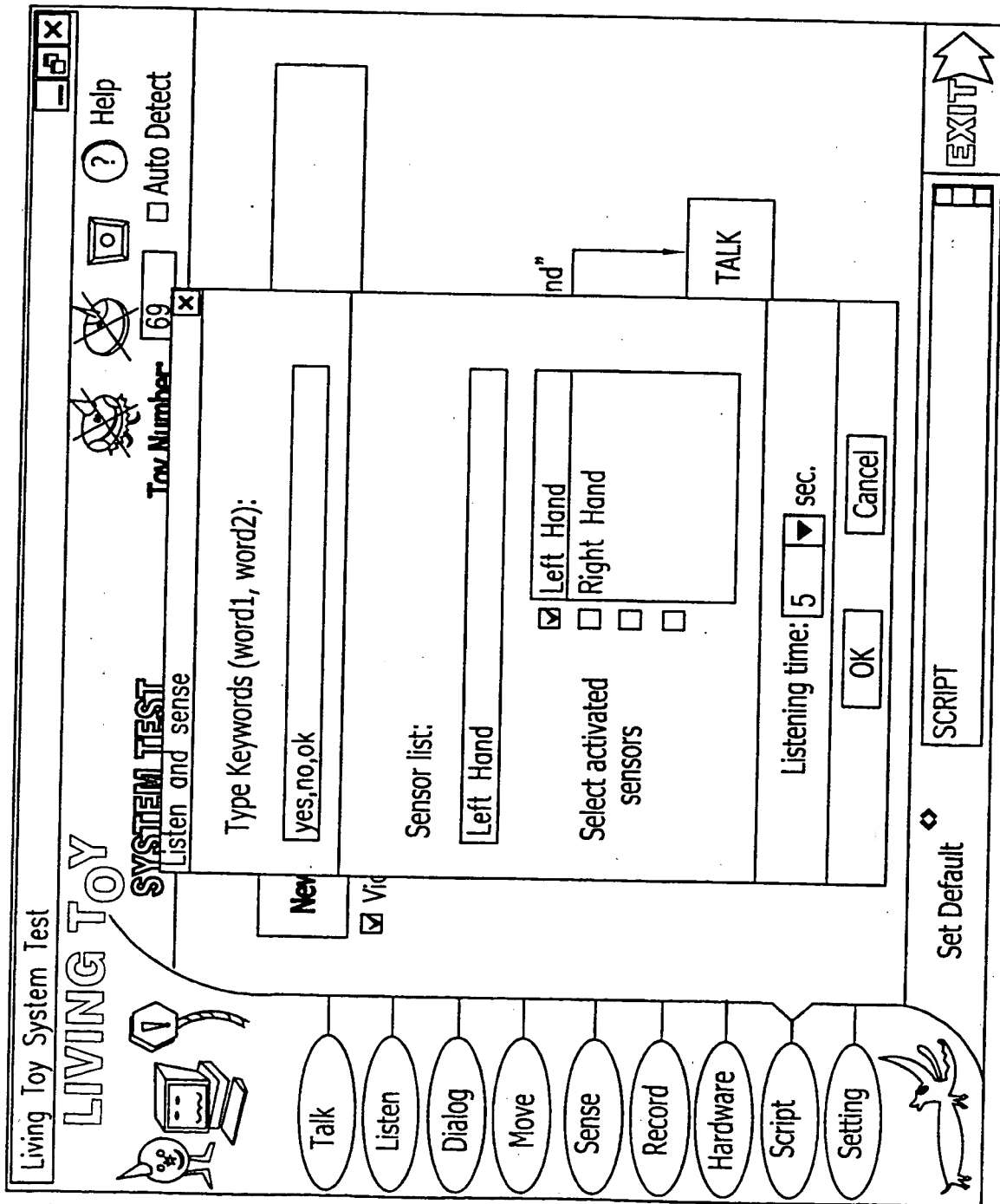


FIG. 44



49/120

FIG. 45

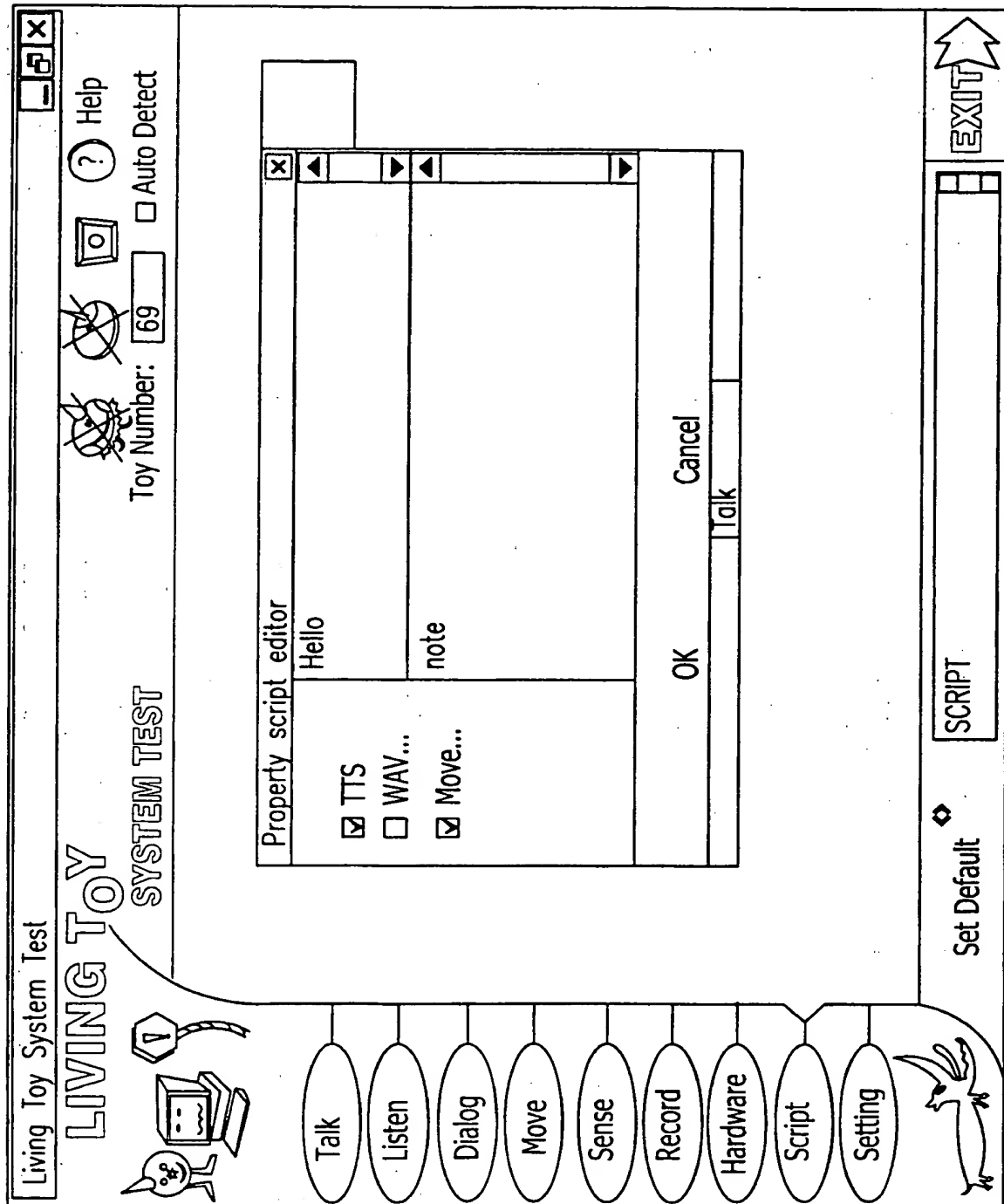


FIG. 46

Living Toy System Test

LIVING TOY

SYSTEM TEST

Toy Number: 69 ☐ Auto Detect

Help ? ☒ ☒ ☒

Johny

Configuration

New OPEN Save As Print Select

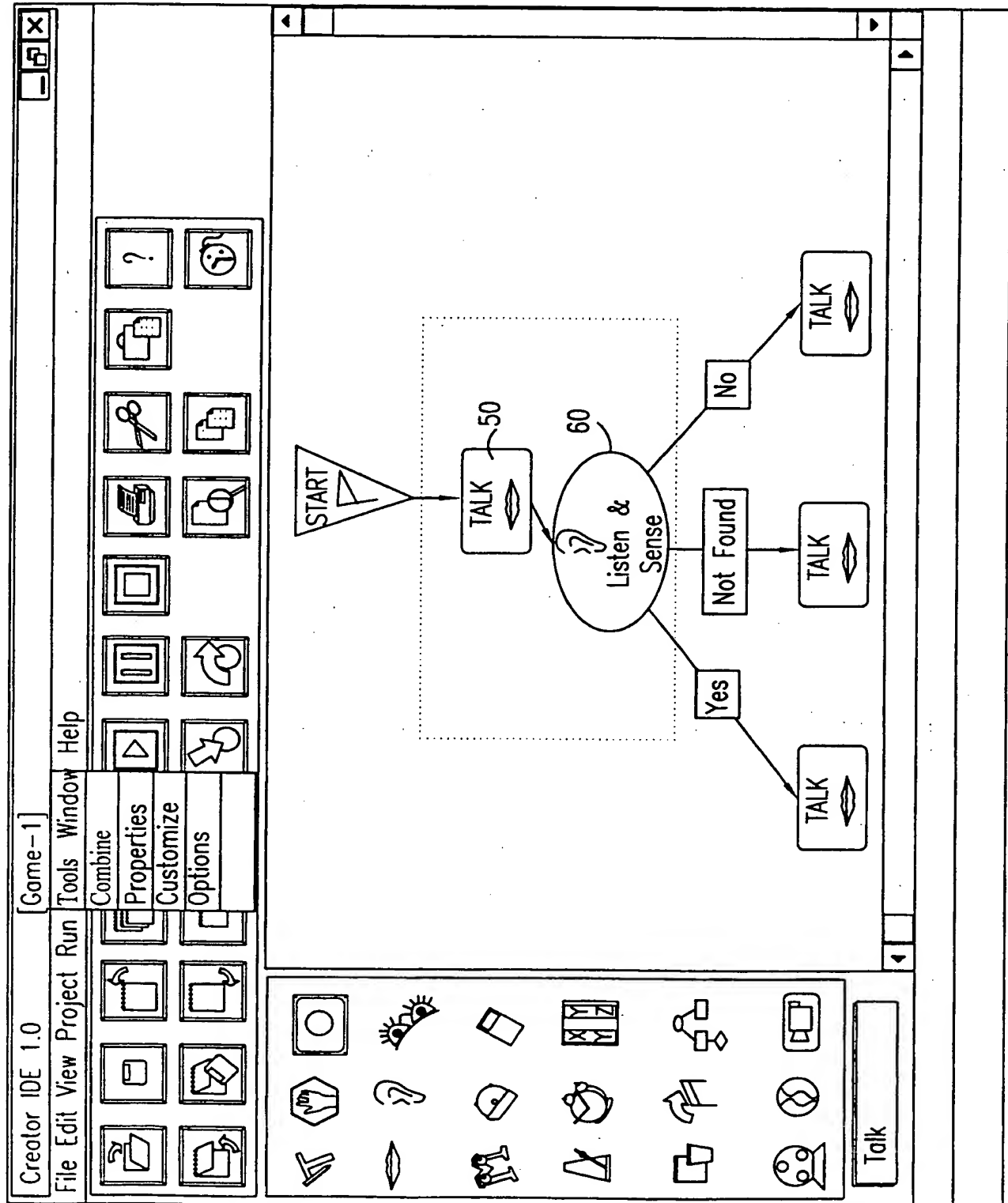
| <u>Movement</u> | | <u>Sensors</u> | |
|-----------------|----------|----------------|-------------|
| Description | Duration | | Description |
| 0 Talk | 0 | 1 | Left Hand |
| 1 Yawn | 2 | 2 | |
| 2 Head Left | 2 | 3 | Right Hand |
| 3 Head Right | 2 | 4 | |
| 4 Head Center | 2 | 5 | |
| 5 Eyes Left | 2 | 6 | |
| 6 Eyes Right | 2 | 7 | |

Talk Listen Dialog Move Sense Record Hardware Script Setting

Set Default **CONFIGURATION** **EXIT**

FIG. 47

51/120



52/120

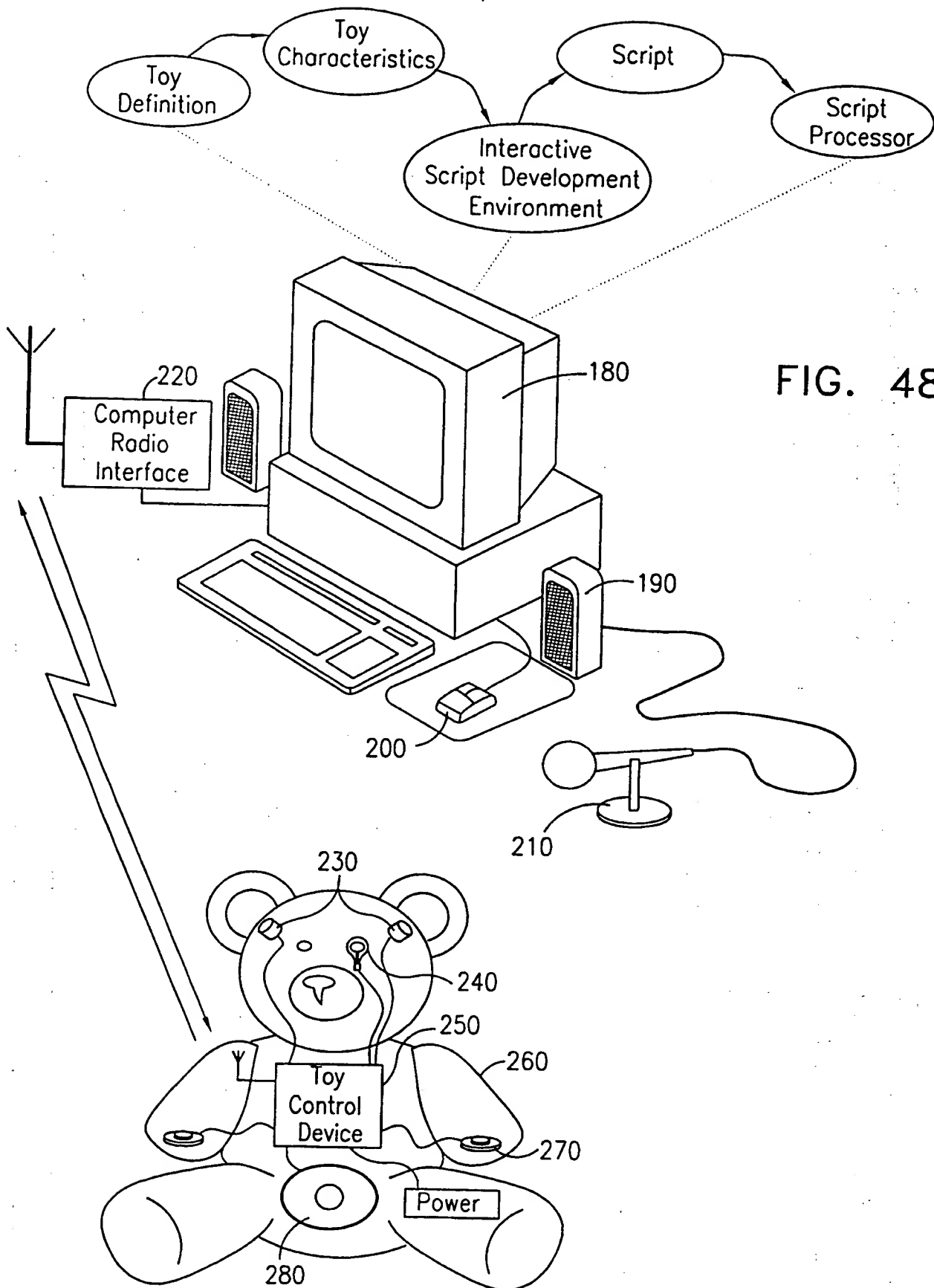
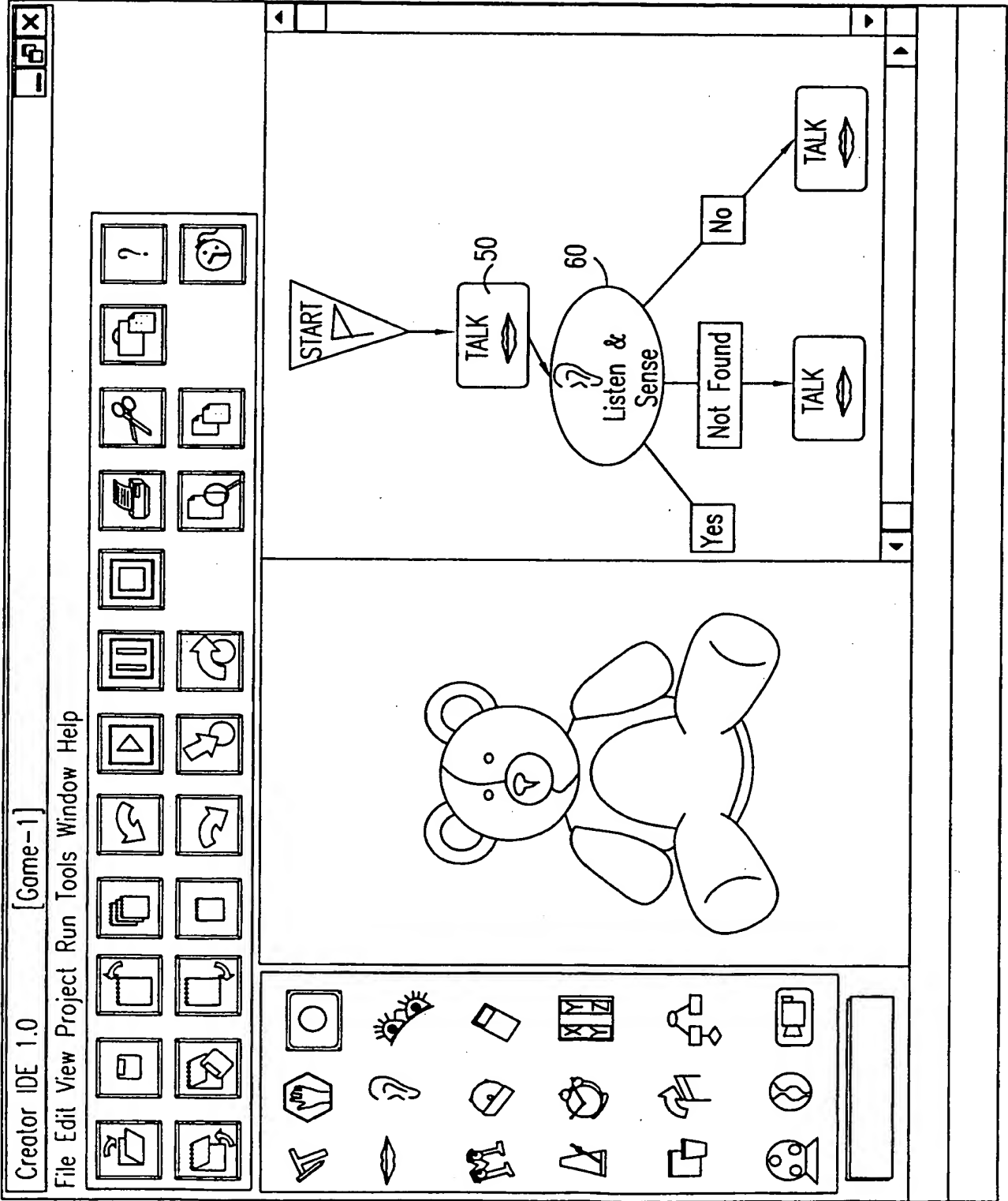


FIG. 48

FIG. 49

53/120



54/120

FIG. 50

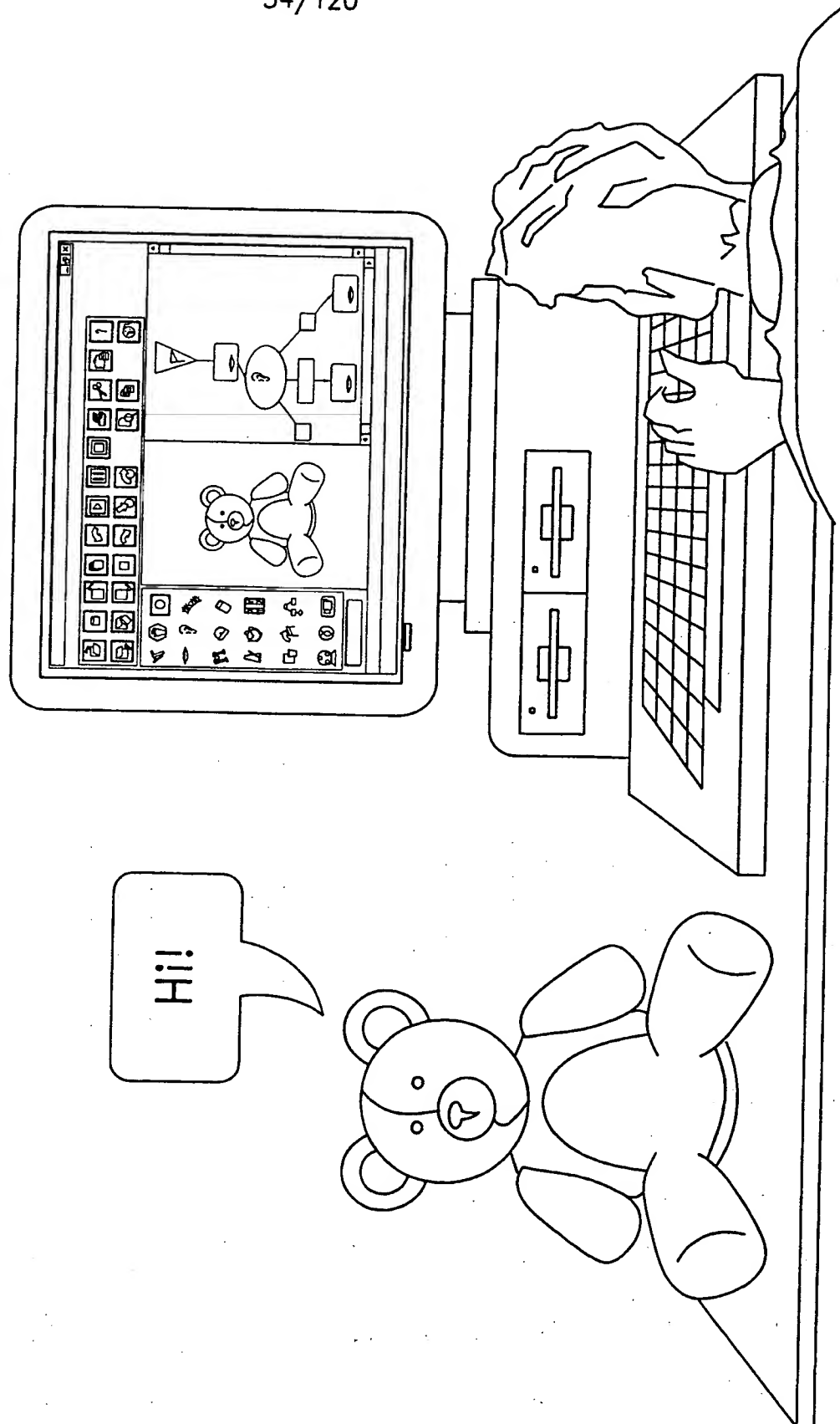
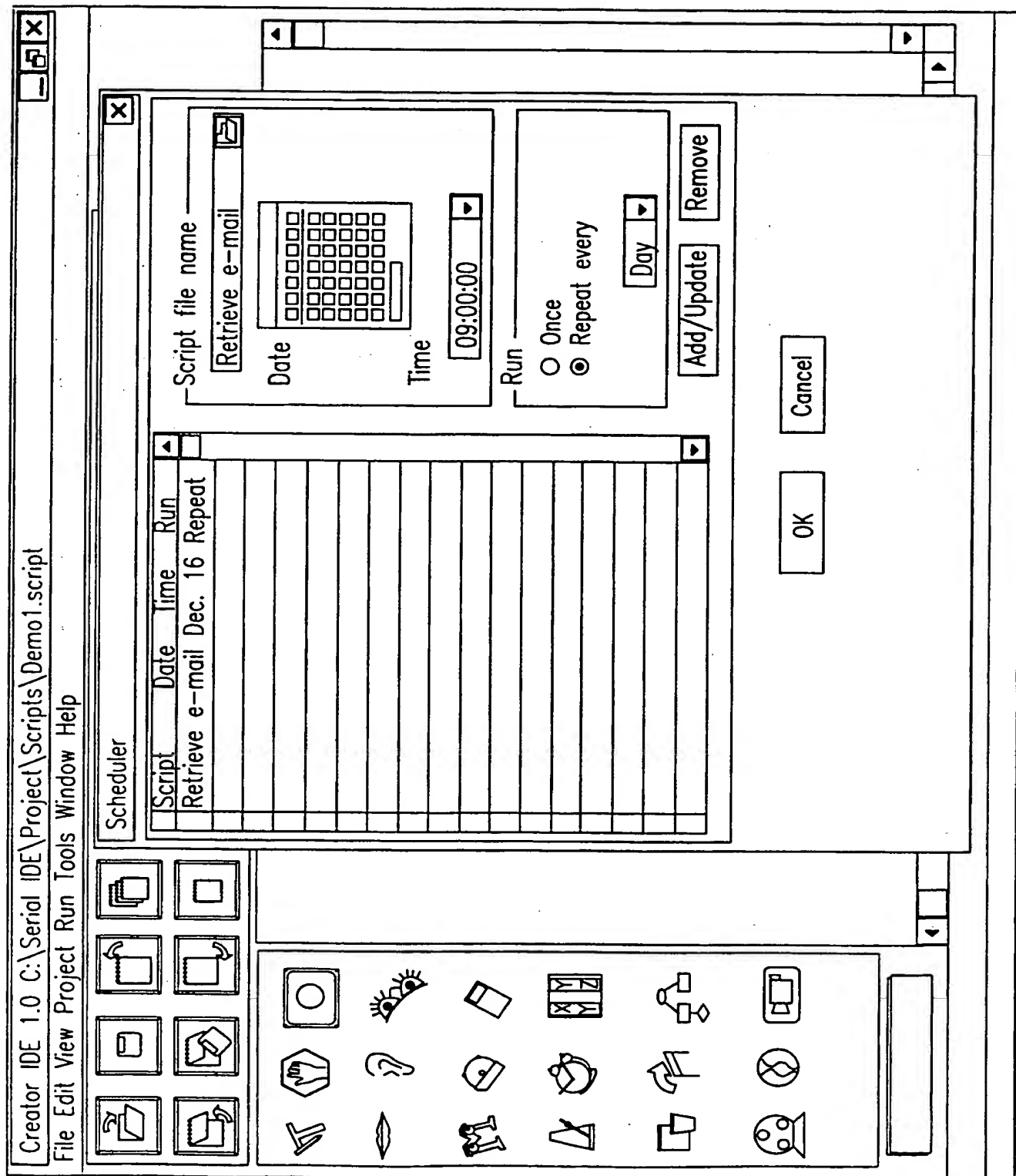
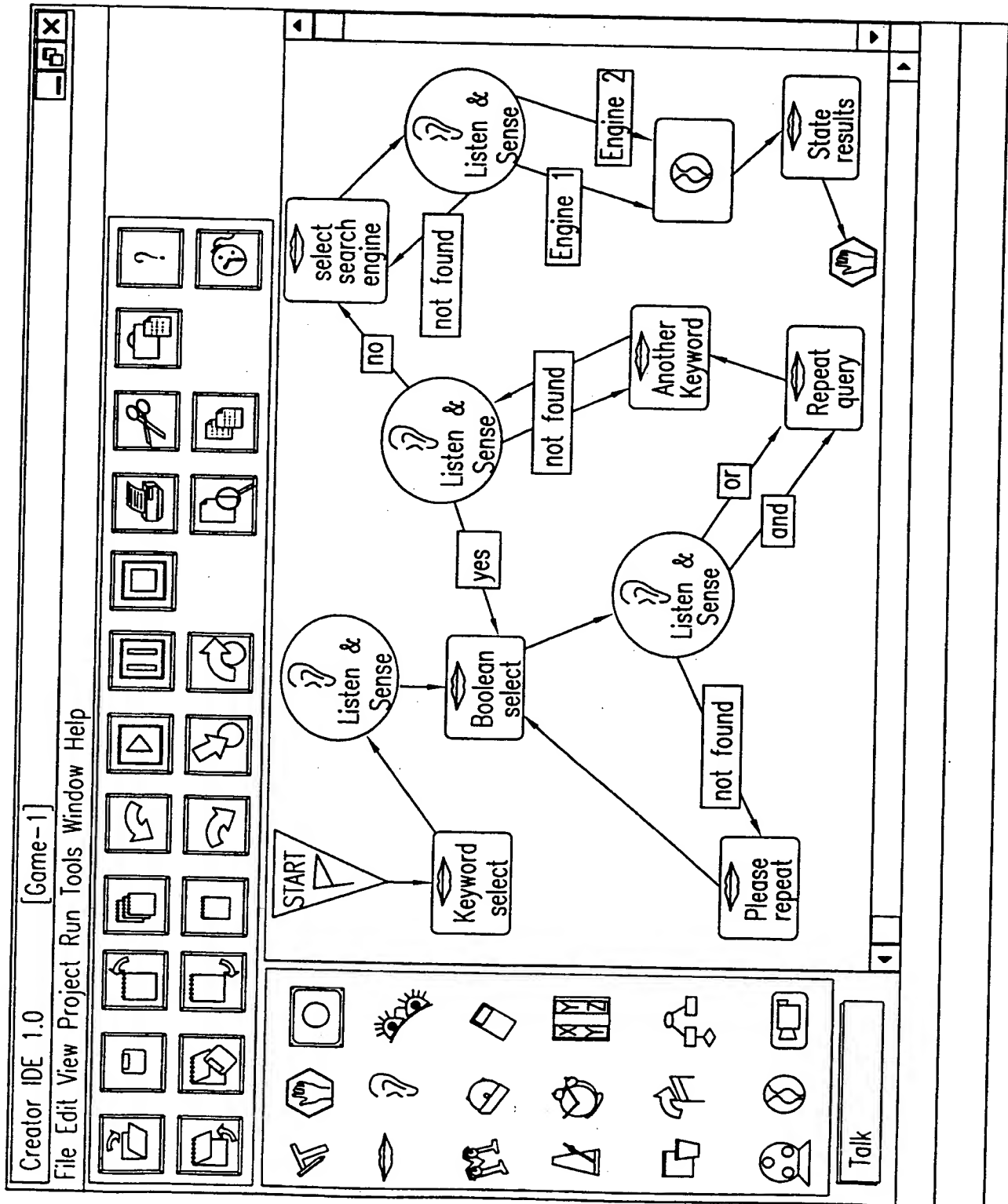


FIG. 51



56/120

FIG. 52



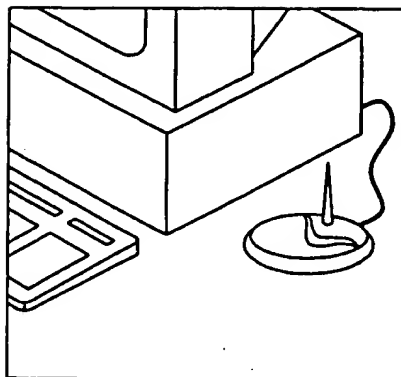


FIG. 53

Living Object Base Station

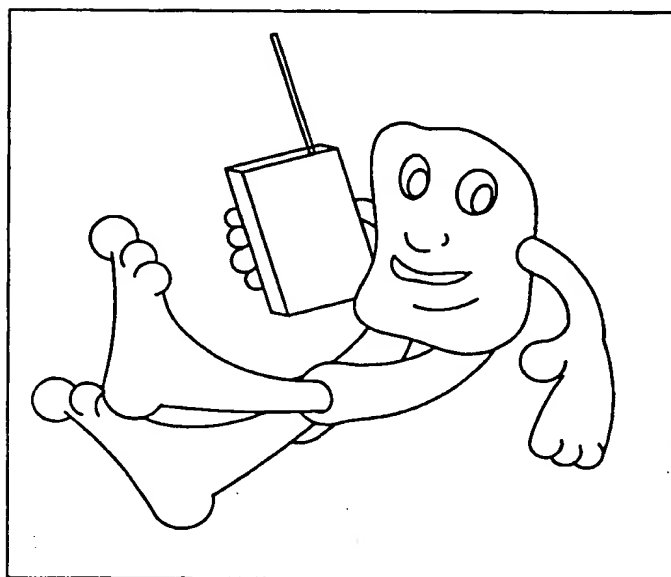


FIG. 54

Living Object Toy

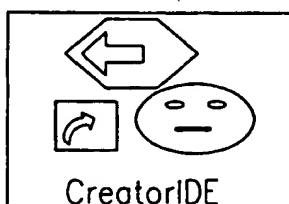
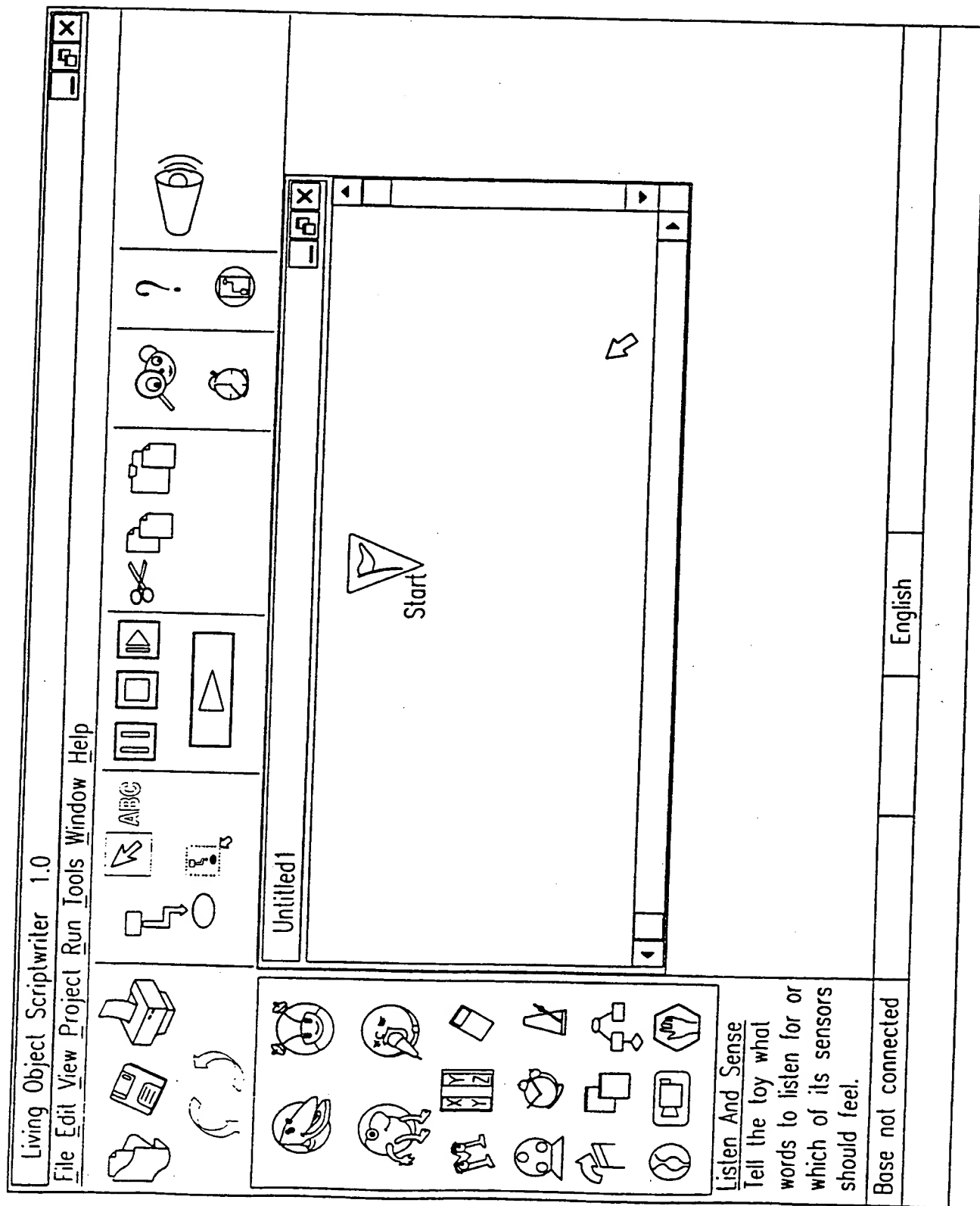


FIG. 55

FIG. 56

58/120



59/120

FIG. 57

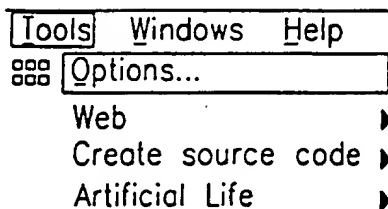
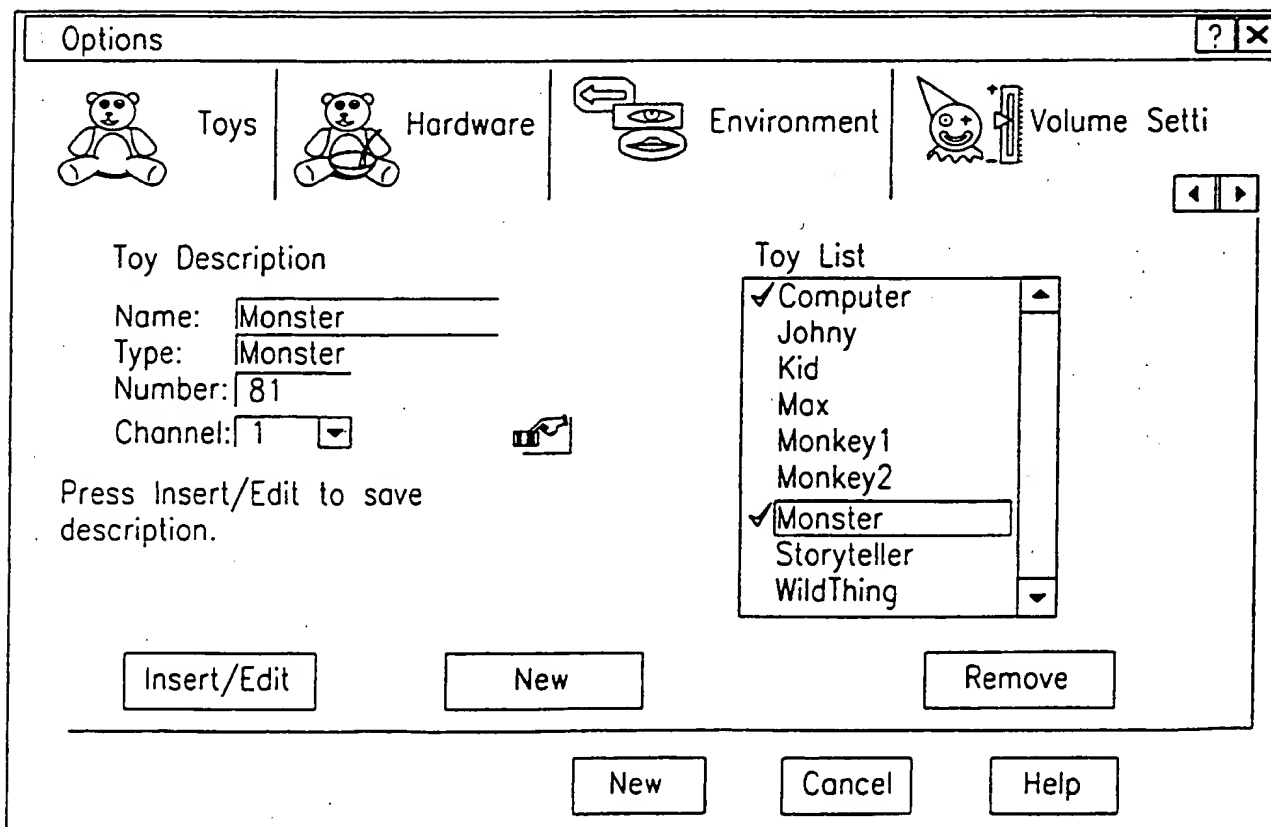


FIG. 58



60/120

FIG. 59

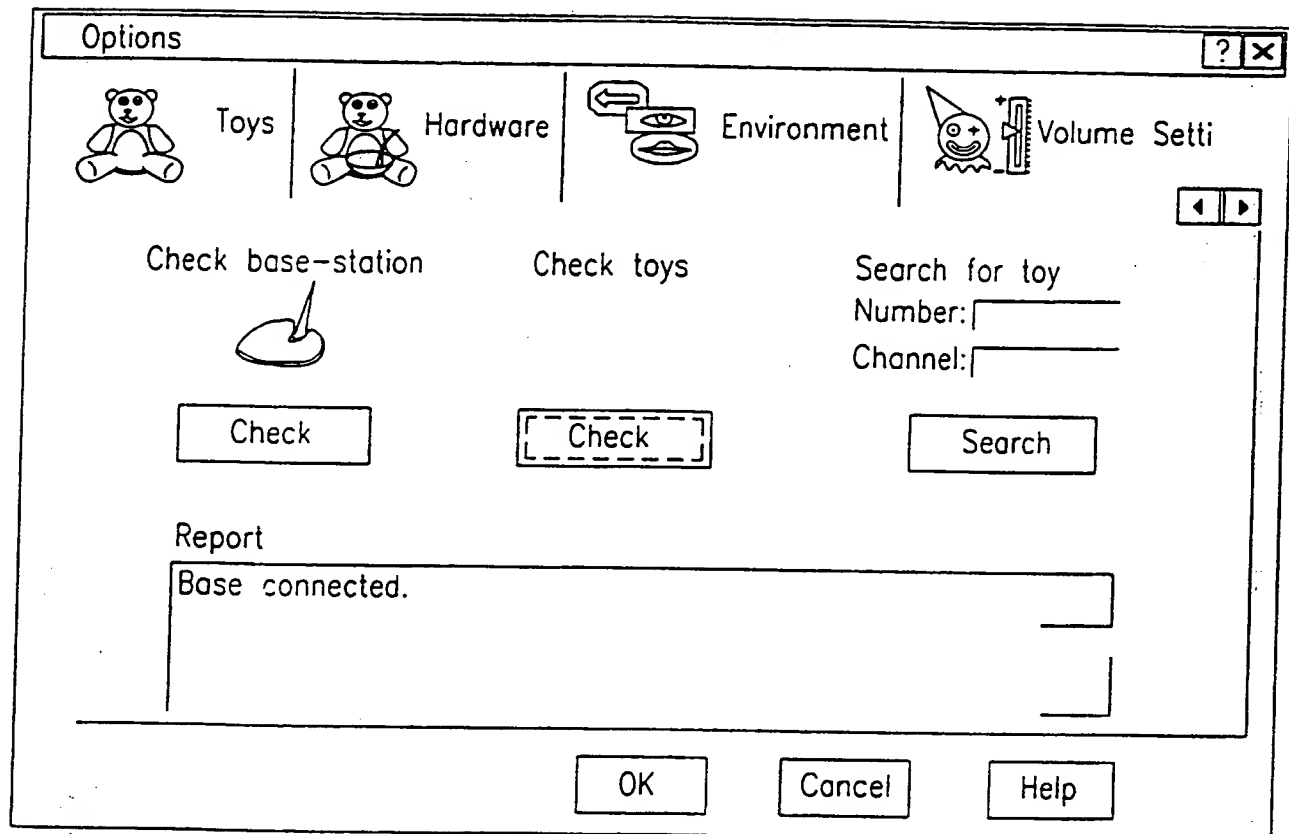
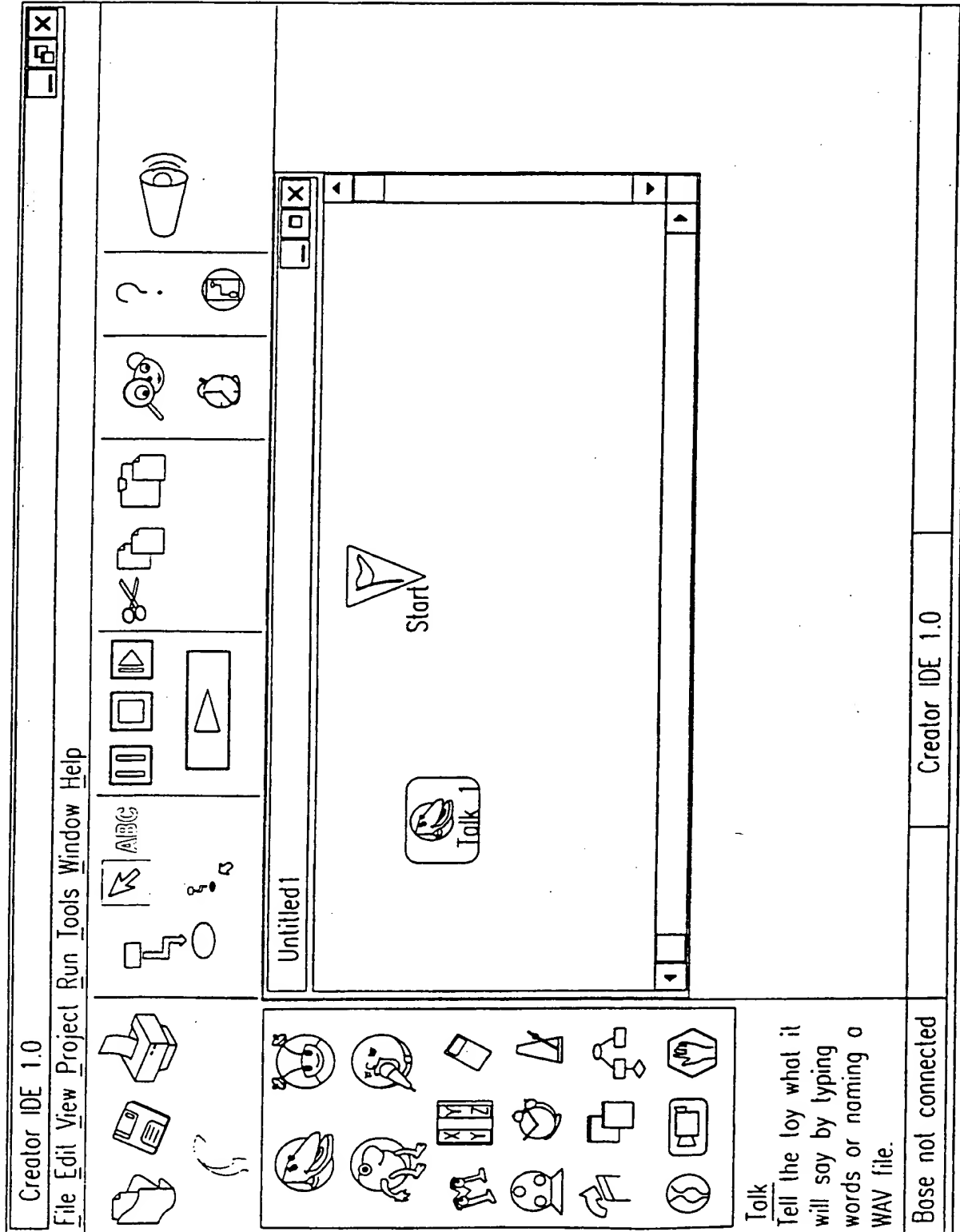


FIG. 60



Talk icon

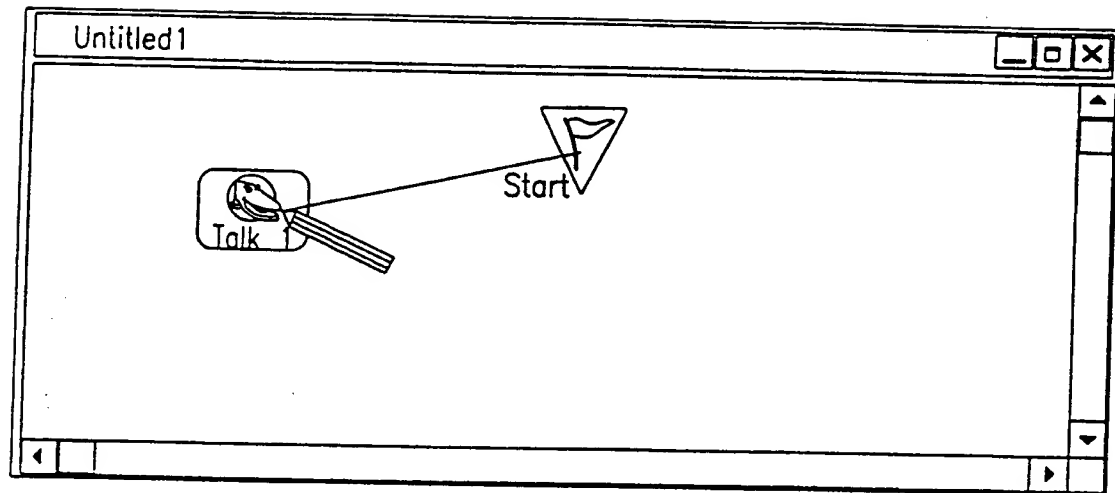
FIG. 61



*Dragging the Talk object
to the desktop*

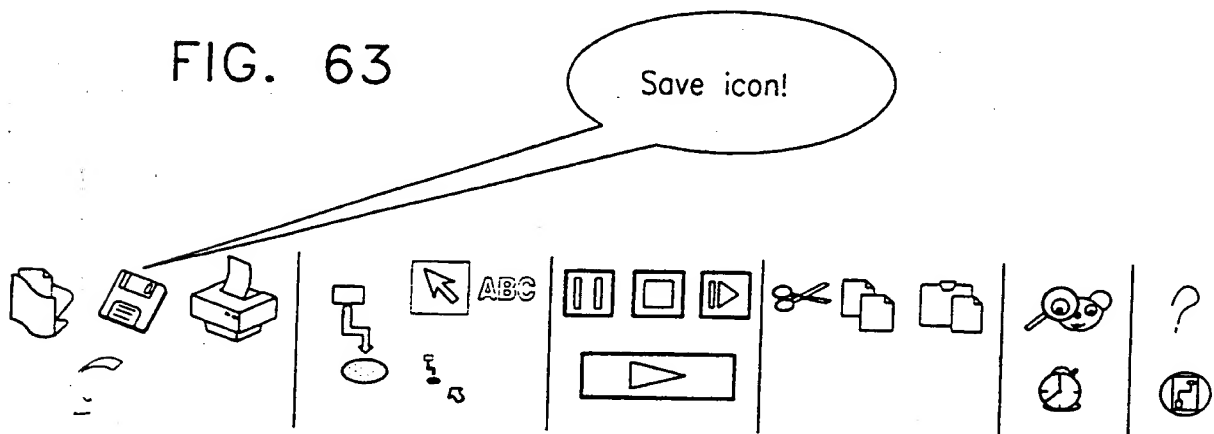
62/120

FIG. 62



Connecting objects

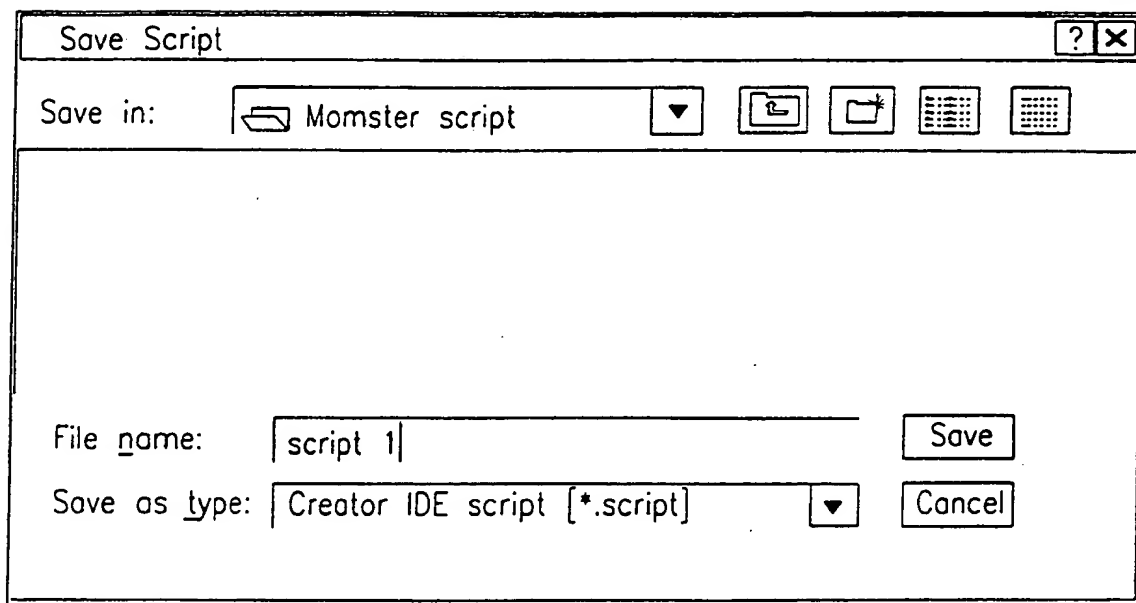
FIG. 63



Actions toolbar: Save icon

63/120

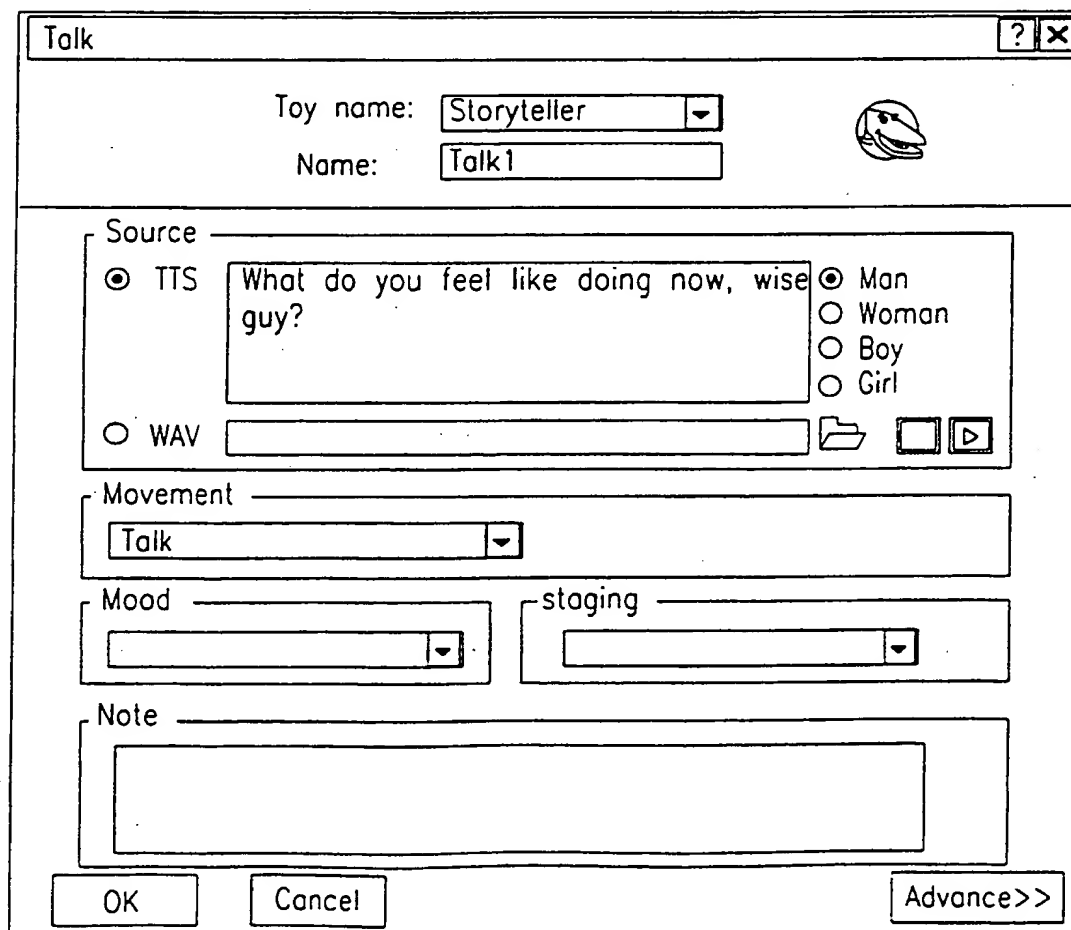
FIG. 64



A dialog box titled "Save Script" with a standard Windows-style title bar (minimize, maximize, close buttons). Below the title bar is a "Save in:" section with a dropdown menu showing "Momster script" and several icons for file operations. The main area of the dialog is empty. At the bottom, there is a "File name:" field containing "script 1", a "Save" button, a "Save as type:" dropdown menu showing "Creator IDE script (*.script)", and a "Cancel" button.

Naming and saving a script

FIG. 65



A dialog box titled "Talk" with a standard Windows-style title bar. It contains several sections: "Toy name:" with a dropdown menu showing "Storyteller" and a "Name:" field showing "Talk1". To the right of these fields is a small icon of a person's head. Below this is a "Source" section with two radio buttons: "TTS" (selected) and "WAV". The "TTS" section has a text field containing "What do you feel like doing now, wise guy?" and a list of gender options: "Man", "Woman", "Boy", and "Girl". The "WAV" section has a file selection icon and a play button icon. Below the "Source" section is a "Movement" section with a dropdown menu showing "Talk". Below that are "Mood" and "staging" sections, each with a dropdown menu. At the bottom is a "Note" section with a large text area. At the very bottom are three buttons: "OK", "Cancel", and "Advance>>".

64/120

FIG. 66

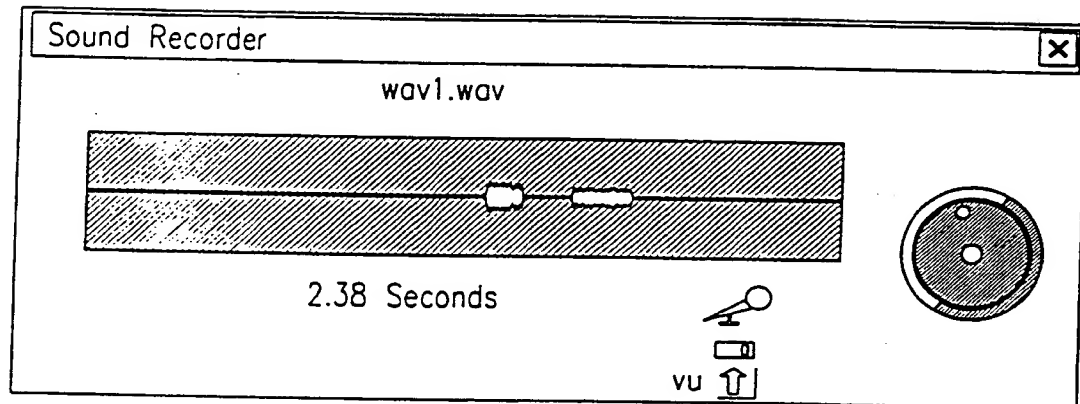
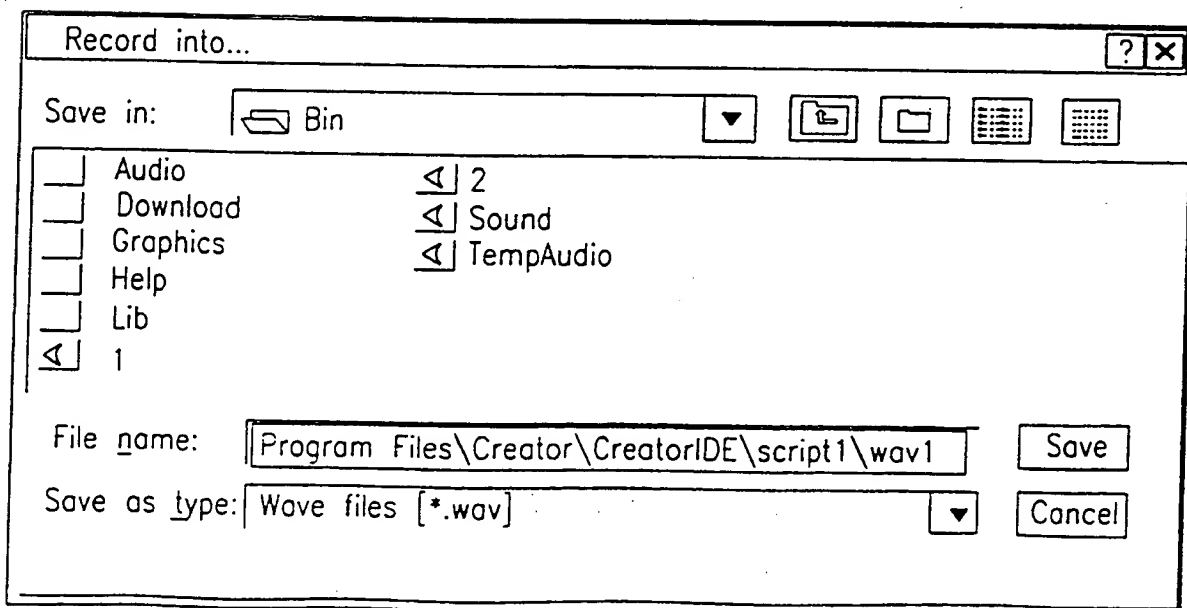


FIG. 67



65/120

FIG. 68

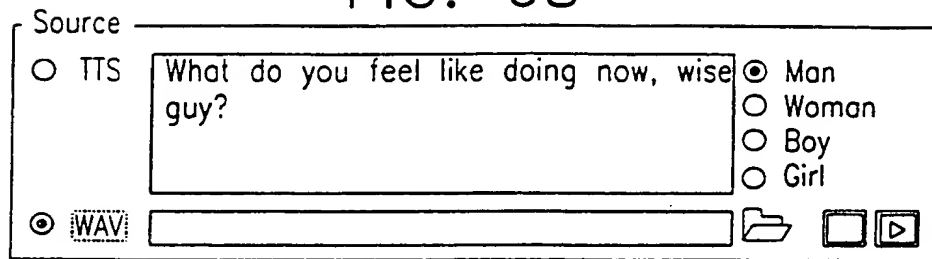
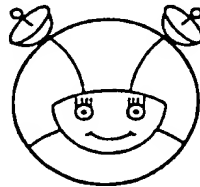
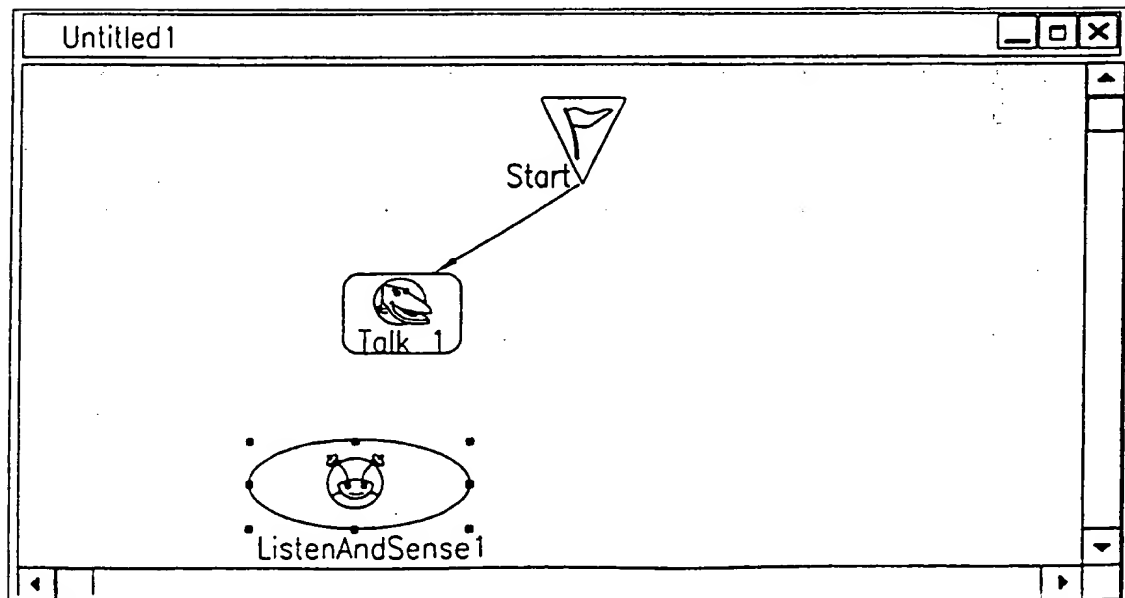


FIG. 69



Listen icon

FIG. 70




Adding the Listen object to your script


66/120

FIG. 71

Listen And Sense [?] [X]

Toy name: 

Name:

Keywords [value1, value2,...]
 

Sensors

| | | |
|----|--------------------------|---|
| #1 | <input type="checkbox"/> | 1 |
| #2 | <input type="checkbox"/> | 2 |
| #3 | <input type="checkbox"/> | 3 |
| #4 | <input type="checkbox"/> | 4 |
| #5 | <input type="checkbox"/> | 5 |
| #6 | <input type="checkbox"/> | 6 |
| #7 | <input type="checkbox"/> | 7 |
| #8 | <input type="checkbox"/> | 8 |

Listen and sense TIME
 sec.

Memory

How many "not-found"

Accuracy
☐ Active

Note

67/120

FIG. 72

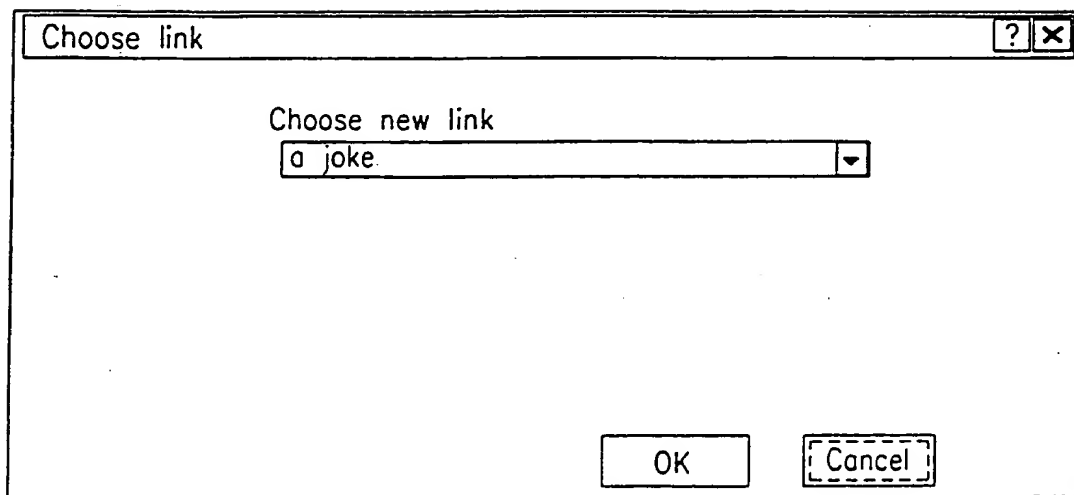
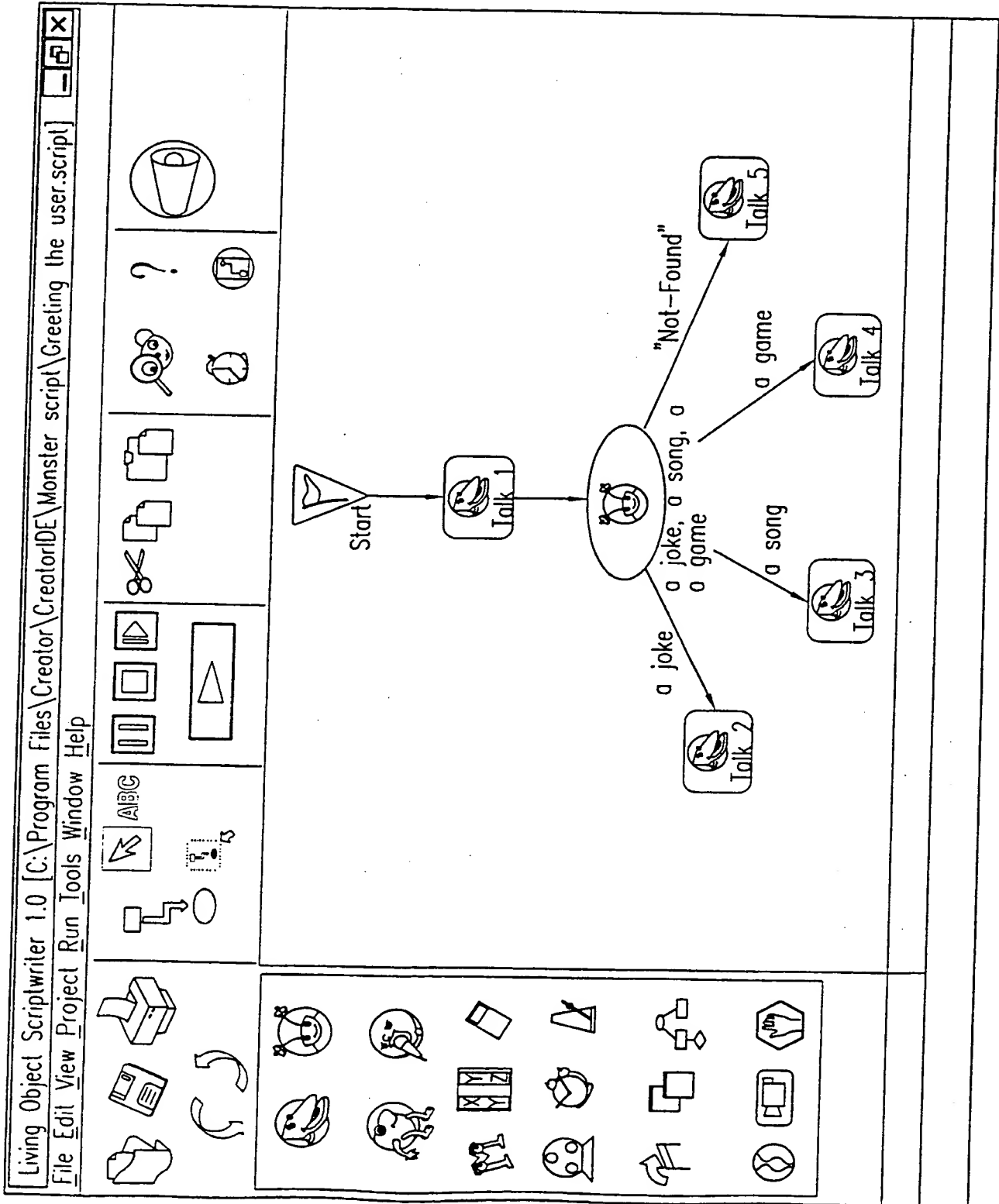


FIG. 73



69/120

FIG. 74

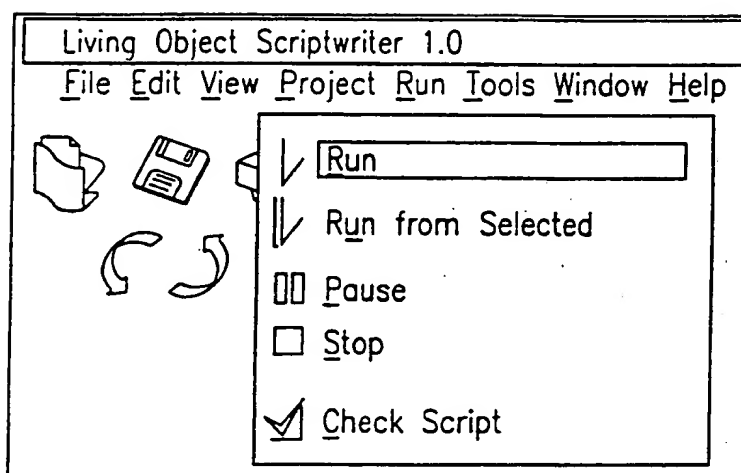
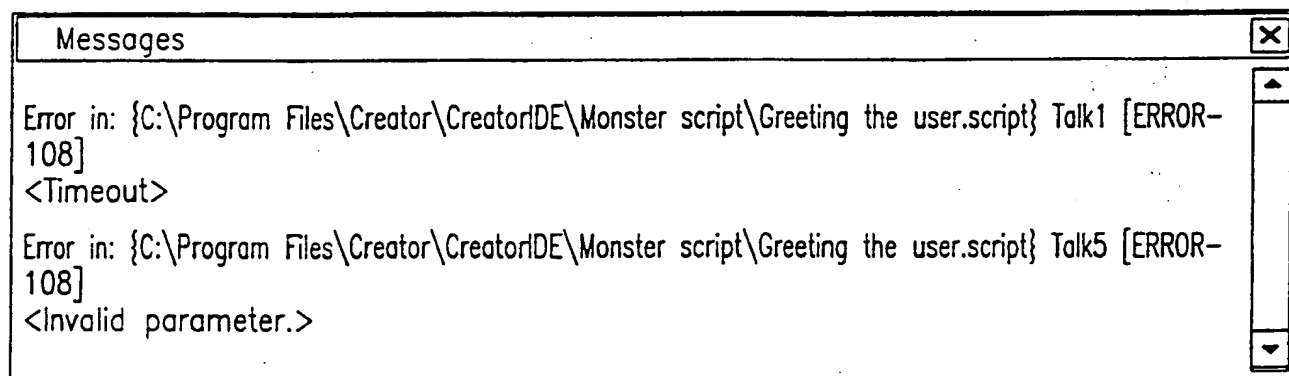





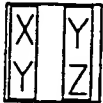



FIG. 75



70/120

FIG. 76A

| | | |
|-------------|---|---|
| Talk |  | Enter speech for the toy |
| Listen |  | Tell the toy to listen and/or to activate (be in a state to receive input from) one of its sensors |
| Move |  | Move one of the toy's moving parts |
| Record |  | Record your voice, your friend's or anyone else's |
| Memory |  | Put a value in a compartment in the computer's memory and give the compartment a name |
| Condition |  | Compare two different values or check if one value is greater than, less than, equal to, or not equal to a certain value. |
| Calculation |  | Do some math. The calculator performs addition, subtraction, multiplication, and division. |

71/120

FIG. 76B

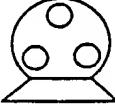




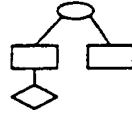



| | | |
|-------------|---|--|
| Random |  | Create a list of values that the computer will choose from on a random basis. Tell the computer in which memory compartment to put the values. |
| Time marker |  | Put a certain time or date in a compartment in the computer's memory |
| Wait |  | Tell the toy to wait for a certain amount of time before proceeding with the script |
| Jump |  | Skip to a different point in the script |
| Execute |  | Run any software program on the computer |
| Script |  | Run any other Scriptwriter script |
| Internet |  | Open a specific page on a particular web site |
| Graphics |  | Show a picture or a video file on the computer's screen |
| End |  | End the script at this point |

FIG. 77

Creator IDE 1.0

File Edit View Project Run Tools Window Help

Talk

Toy name: Storyteller

Name: Talk1

Source

☒ TTS

☐ WAV

say something

Man ☒ Woman ☐ Boy ☐ Girl ☐

Movement Talk

Mood happy

Note

OK Cancel Advance>>

Unit

Talk

Tell the toy what it will say by typing words or naming a WAV file.

Base not connected English

73/120

FIG. 78

Living Object Scriptwriter 1.0

File Edit View Project Run Tools Window Help

Listen And Sense

Toy name: Storyteller

Name: [ListenAndSense1]

Keywords [value1, value2,...]

yes,no

Sensors

#1 ☐ Left Hand

#2 ☐ Nose

#3 ☐ Right Hand

#4 ☐ Foot

#5 ☐

#6 ☐

#7 ☐

#8 ☐

Listen and sence1 TIME

5

Sec.

Memory

How many "not-found"

1

Accuracy

%

Active

☒

Note

OK

Cancel

Advance>>

Untitled1

Listen And Sense

Tell the toy what words to listen for which of its sensors should feel.

Base not connected

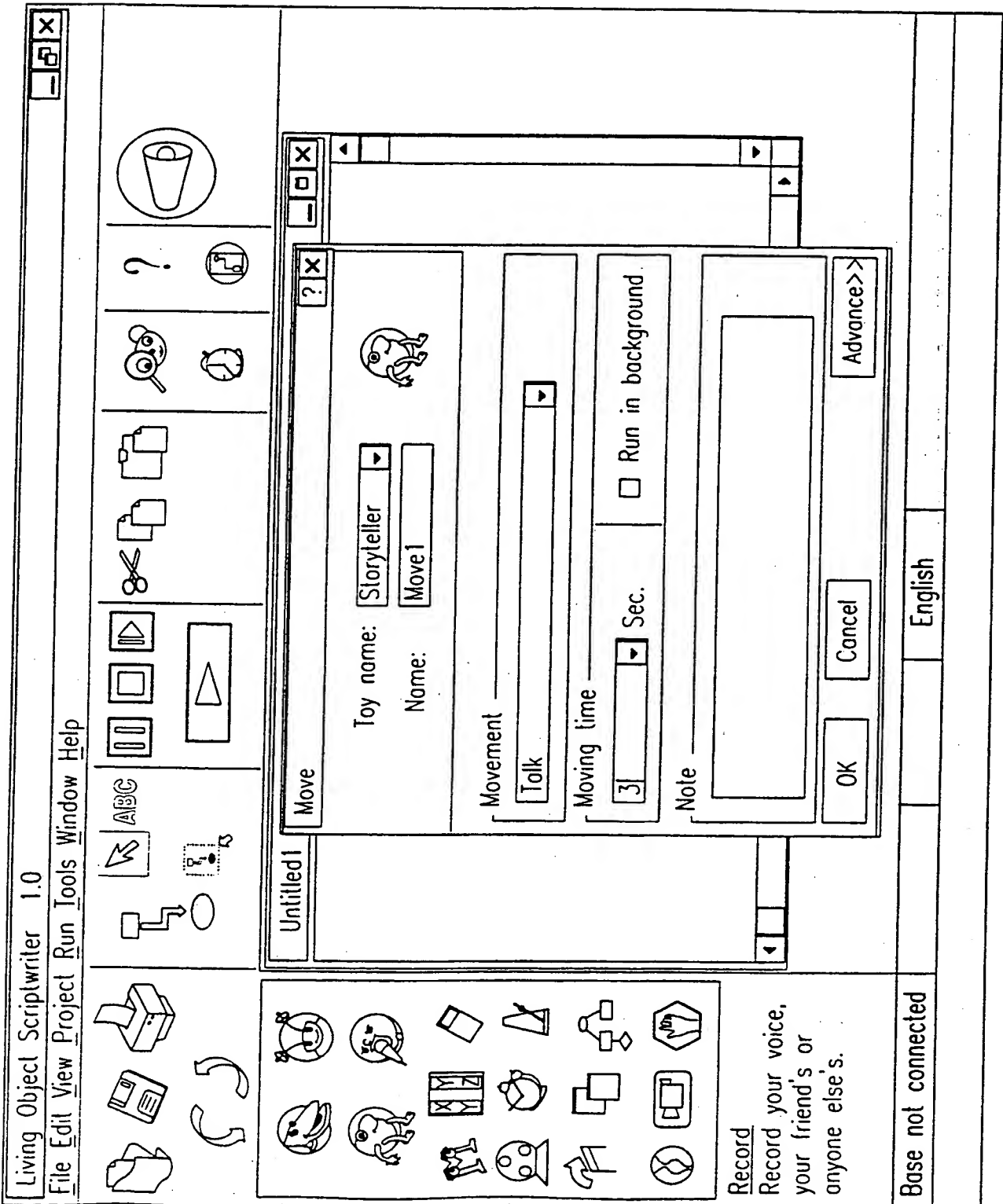
English

SUBSTITUTE SHEET (RULE 26)

ENSDOCID: <WO_0031613A1_1>

74/120

FIG. 79



75/120

FIG. 80

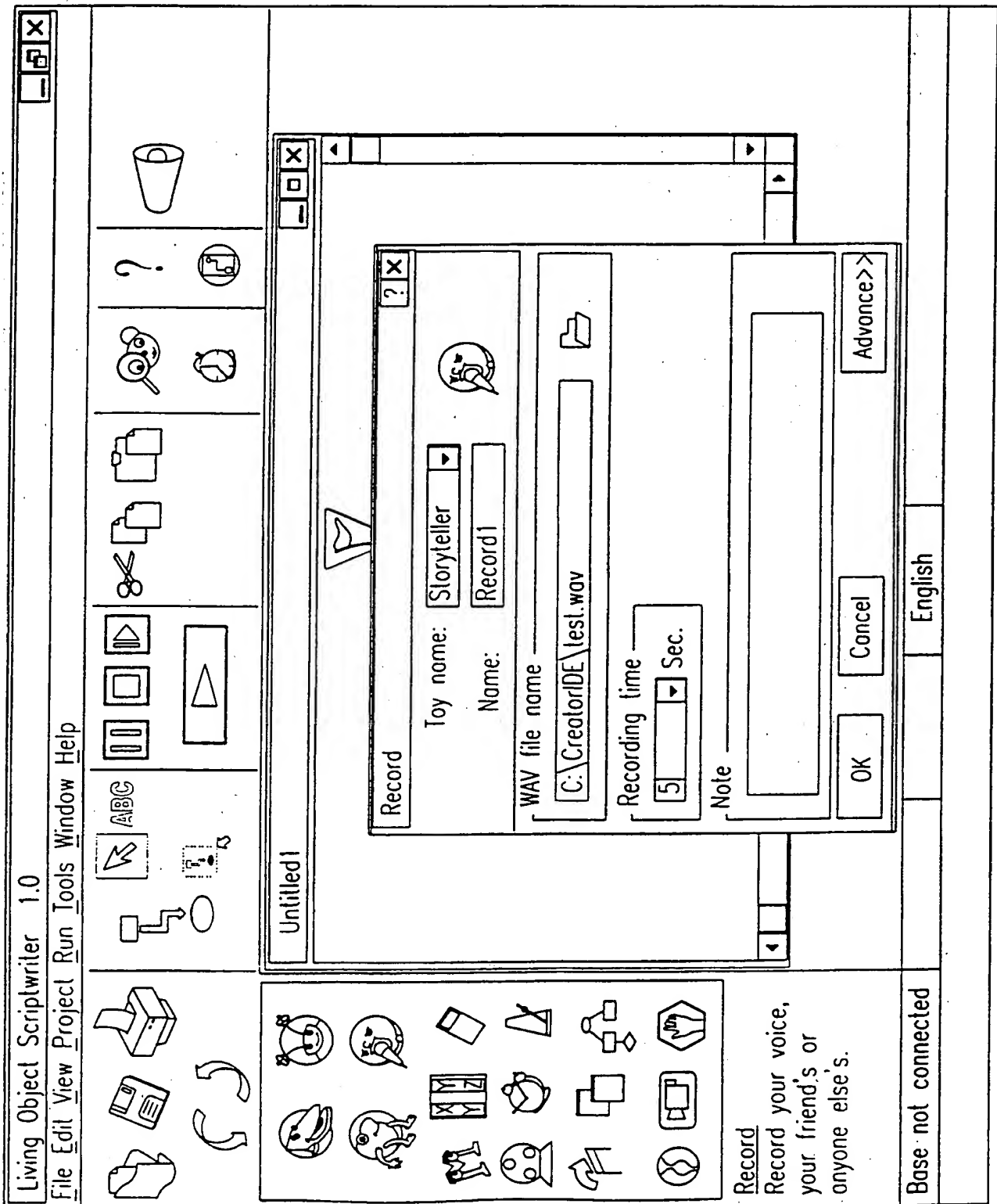


FIG. 81

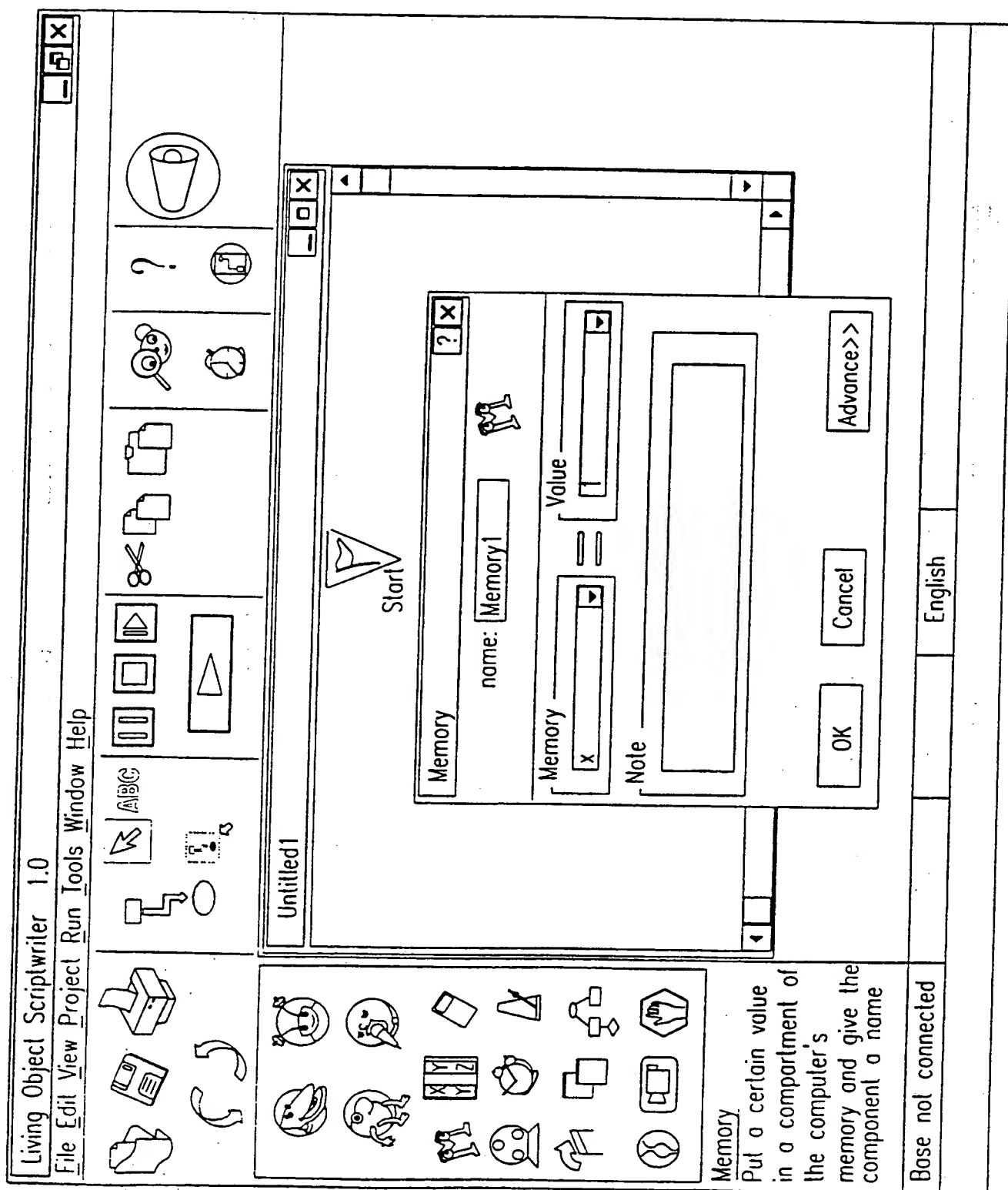
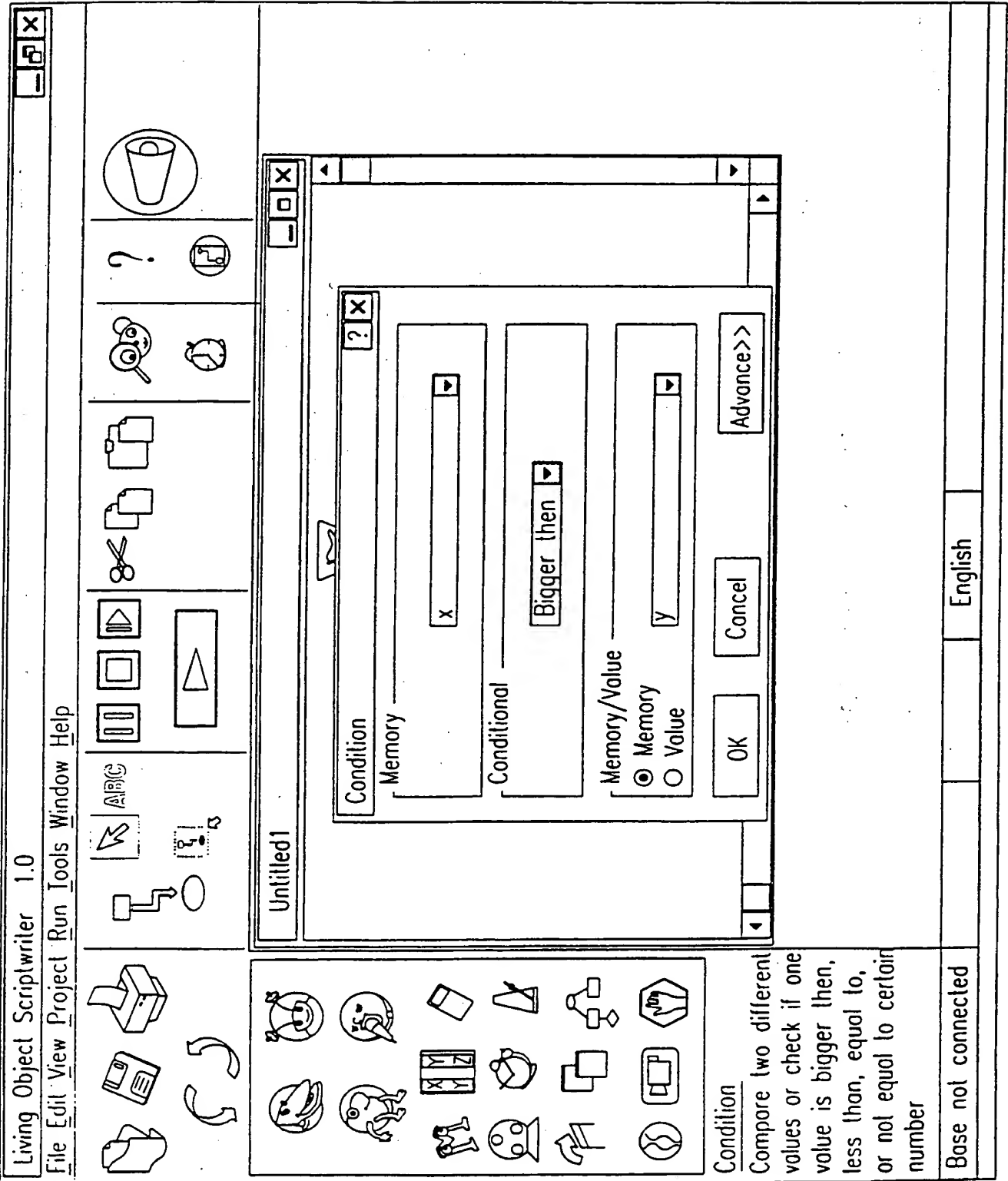
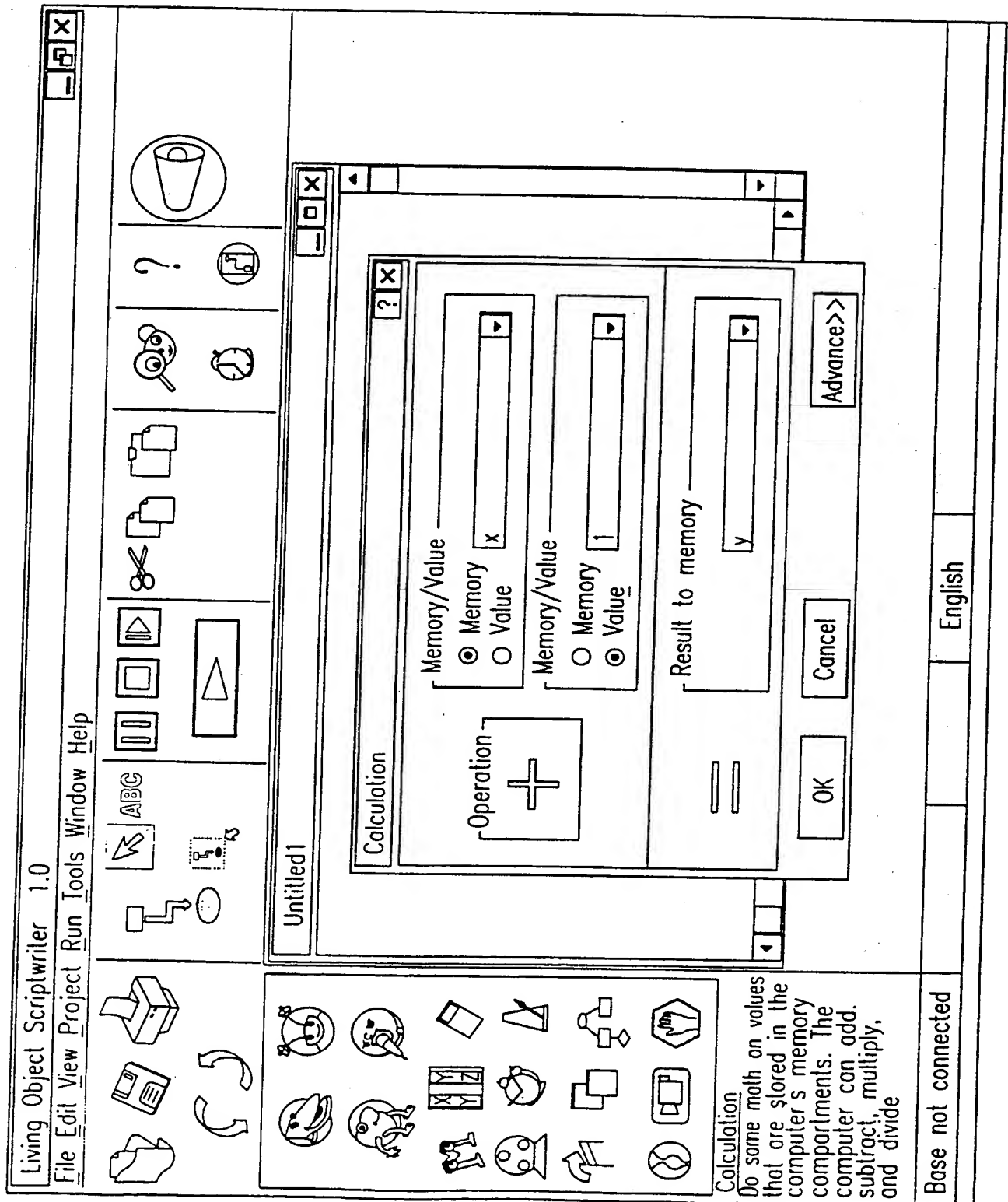


FIG. 82



78/120

FIG. 83



79/120

FIG. 84

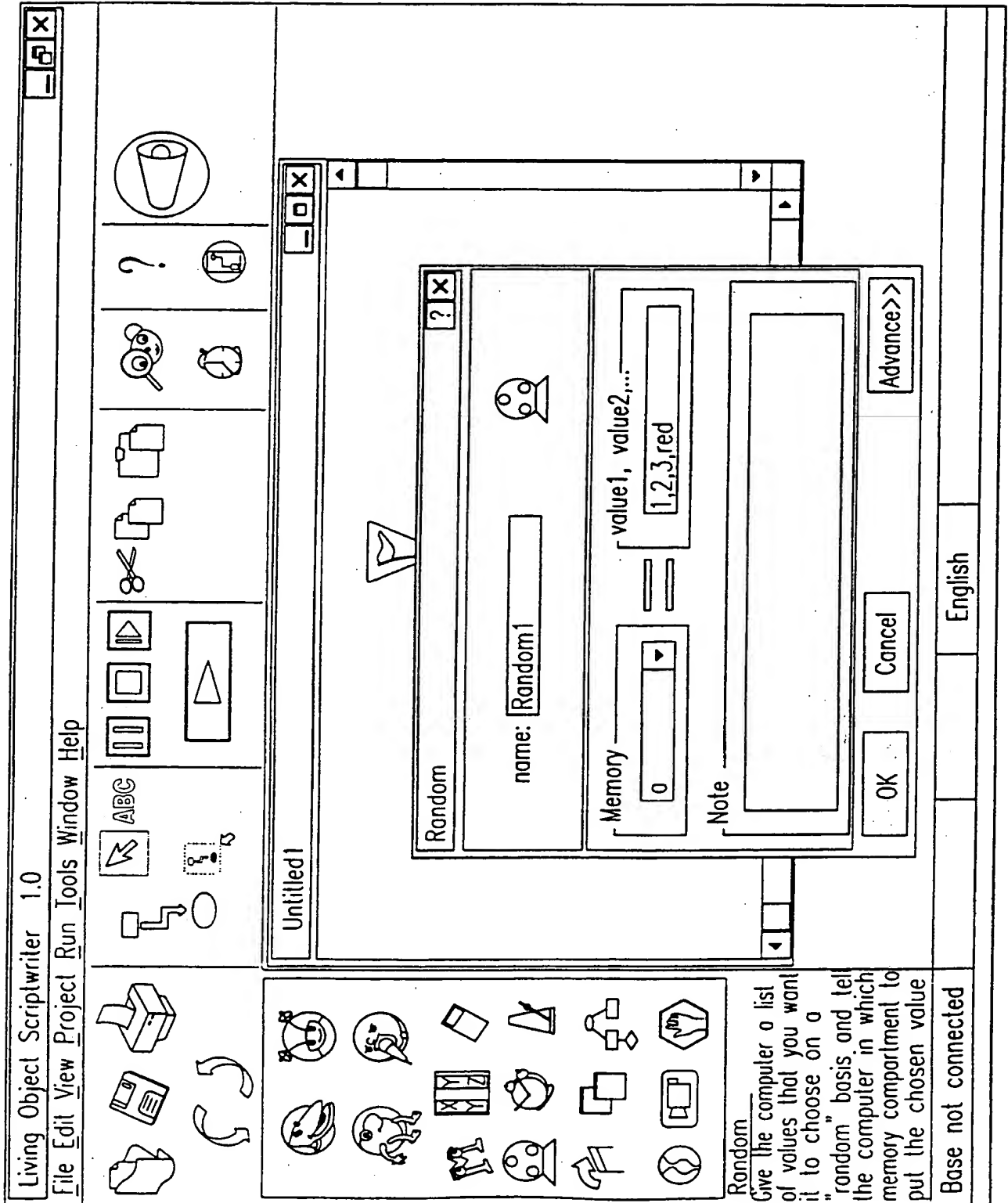


FIG. 85

Living Object Scriptwriter 1.0

File Edit View Project Run Tools Window Help

ABC

name: TimeMarker1

Memory

birthday

Time/Date

☐ Current Time
 ☐ Specific Time

☐ Current Date
 ☐ Specific Date

☐ Current Date/Time
 ☒ Specific Date/Time

Time

15:19:36

Date

12/22/67

Note

OK

Cancel

Advance>>

Time marker

Put a certain time or date (like your birthday), or both, in a compartment of the computer's memory.

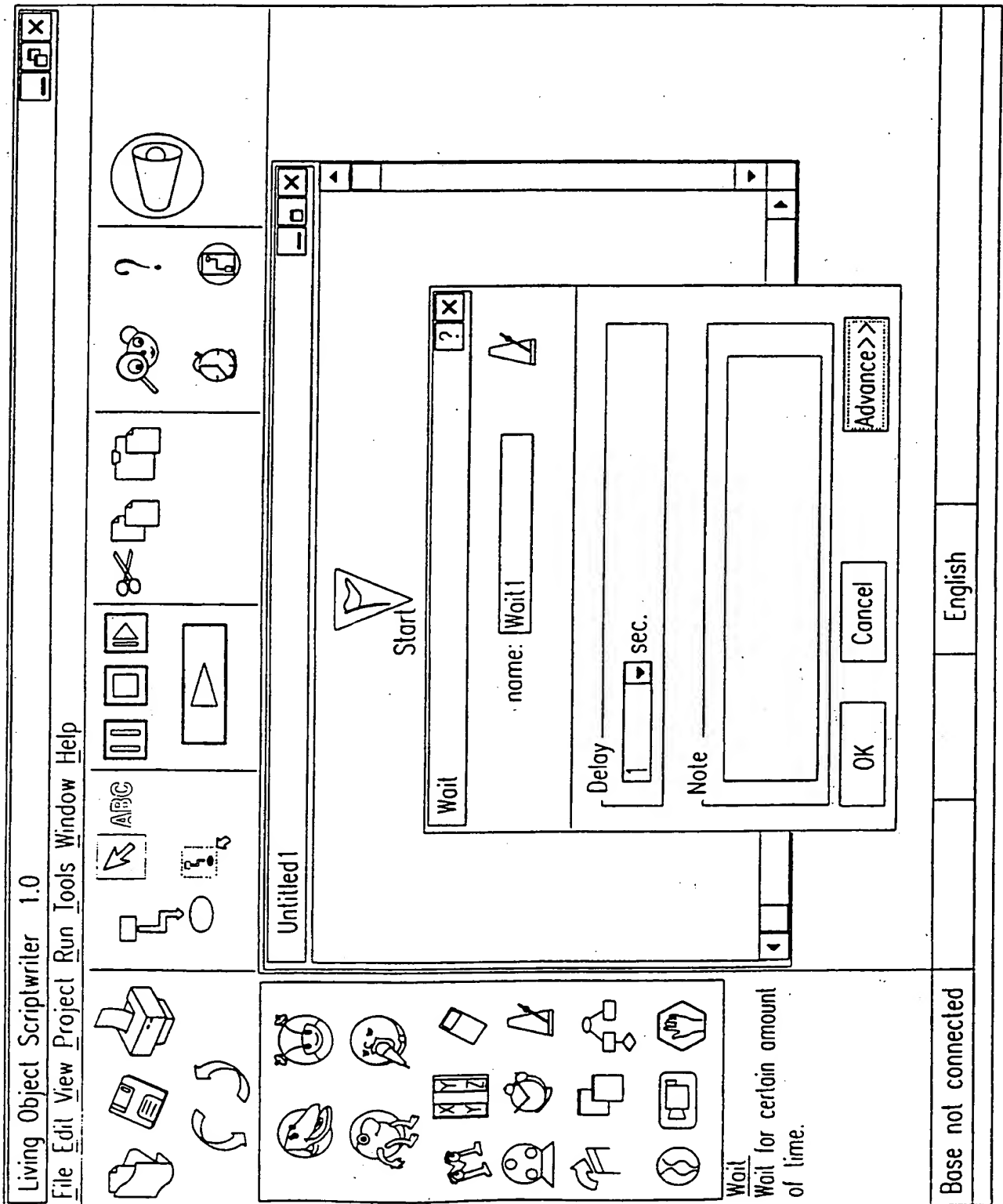
Base not connected

English

ENSDOCID: <WO__0031613A1_1_>

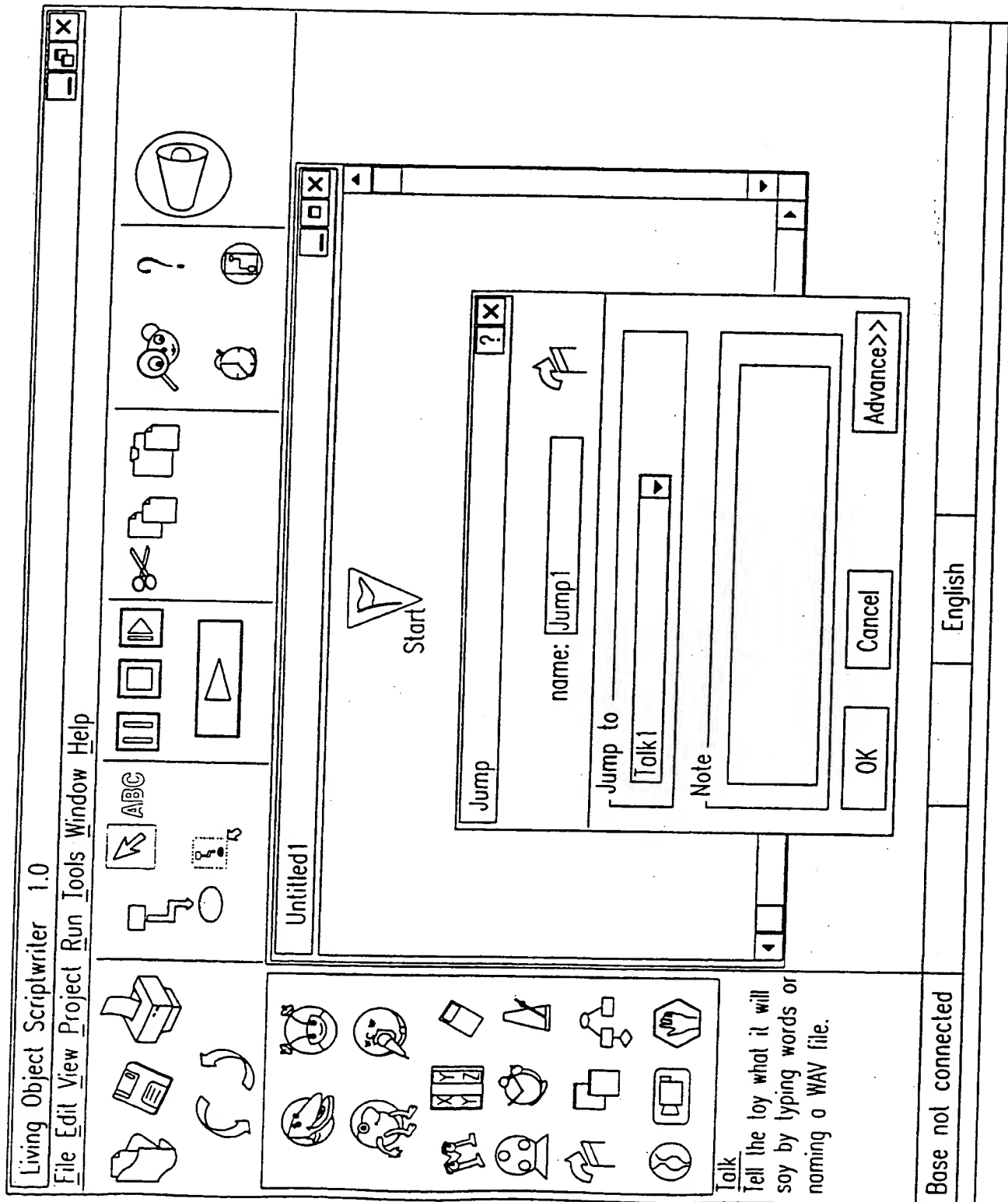
SUBSTITUTE SHEET (RULE 26)

FIG. 86



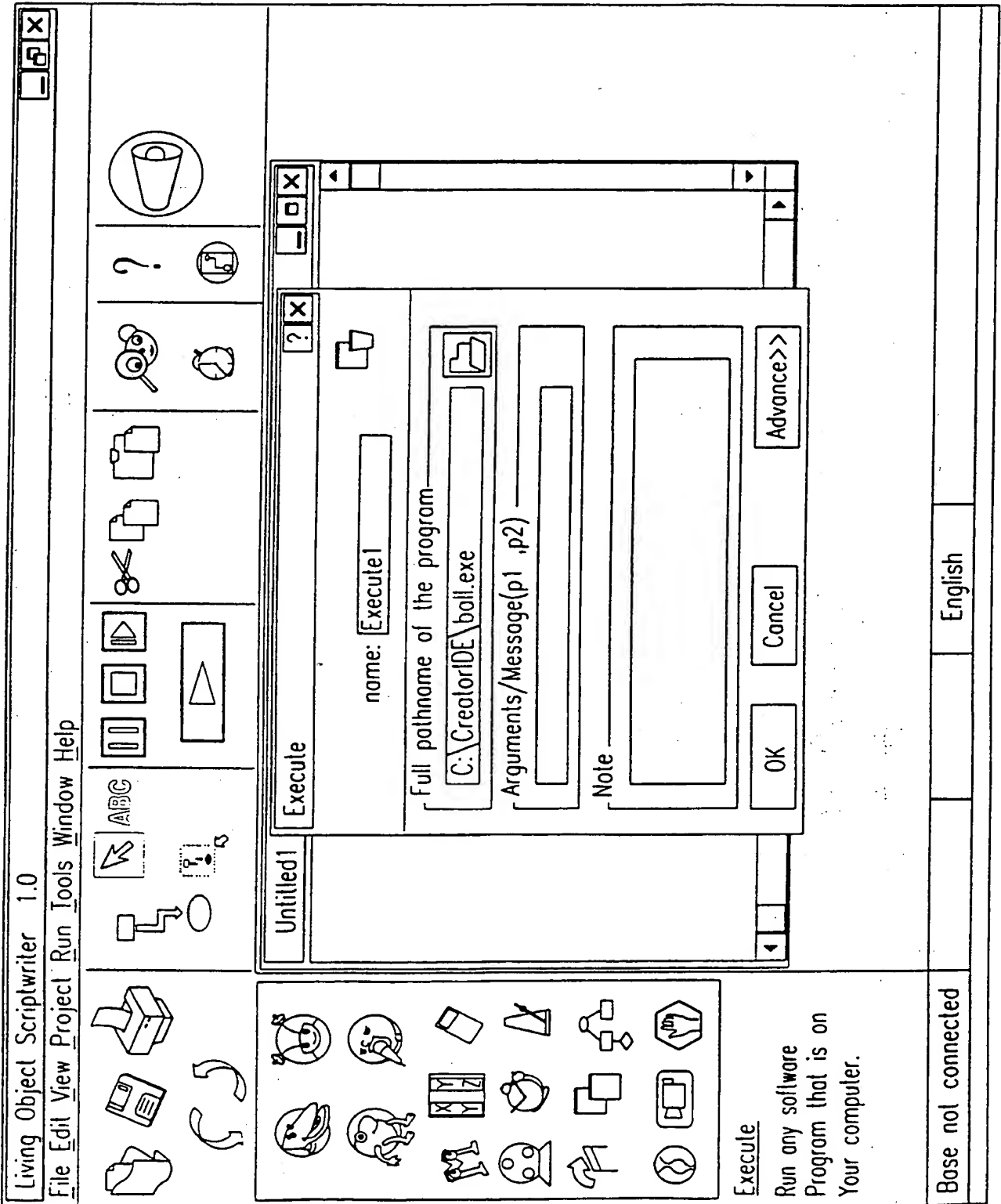
82/120

FIG. 87



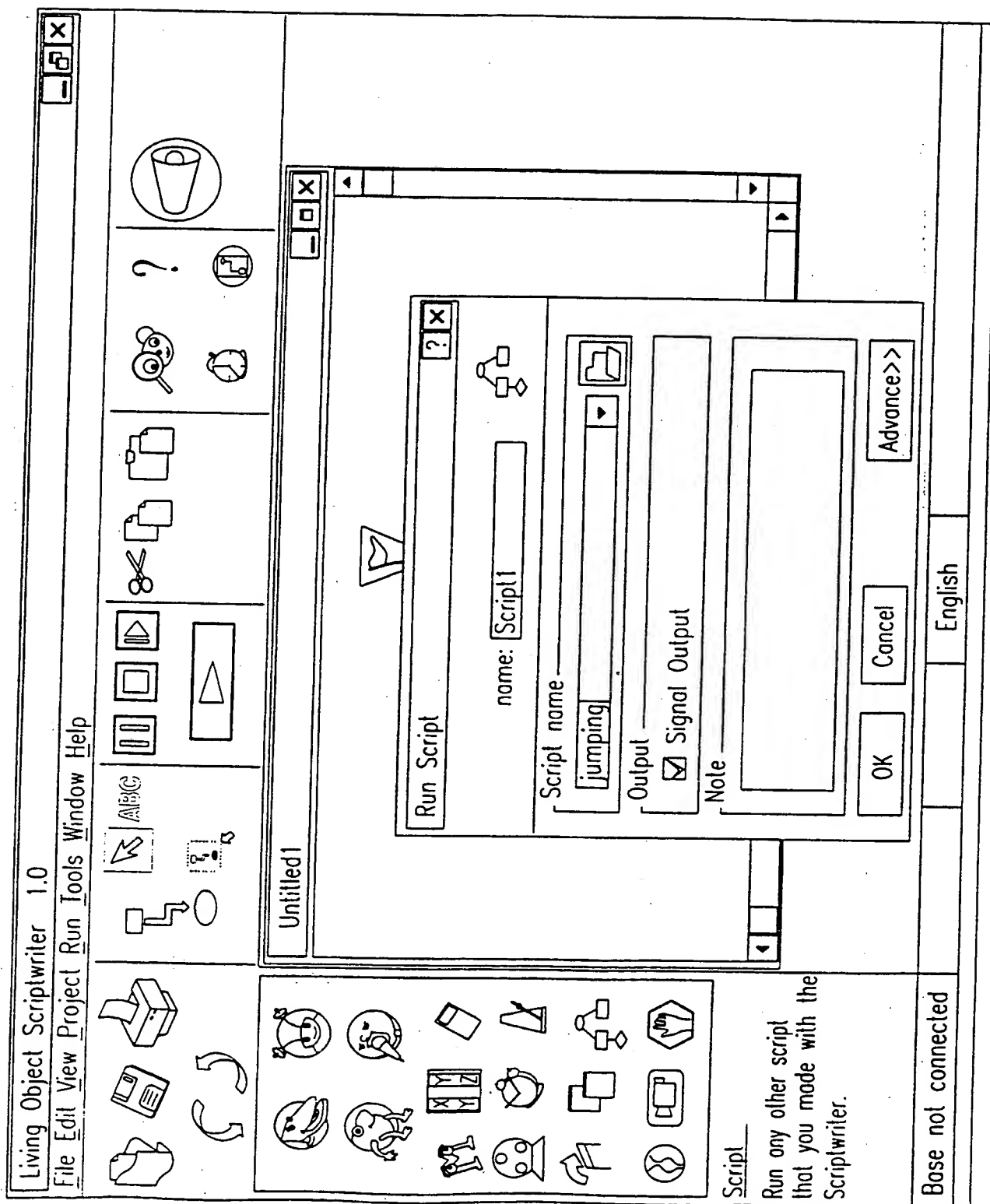
83/120

FIG. 88



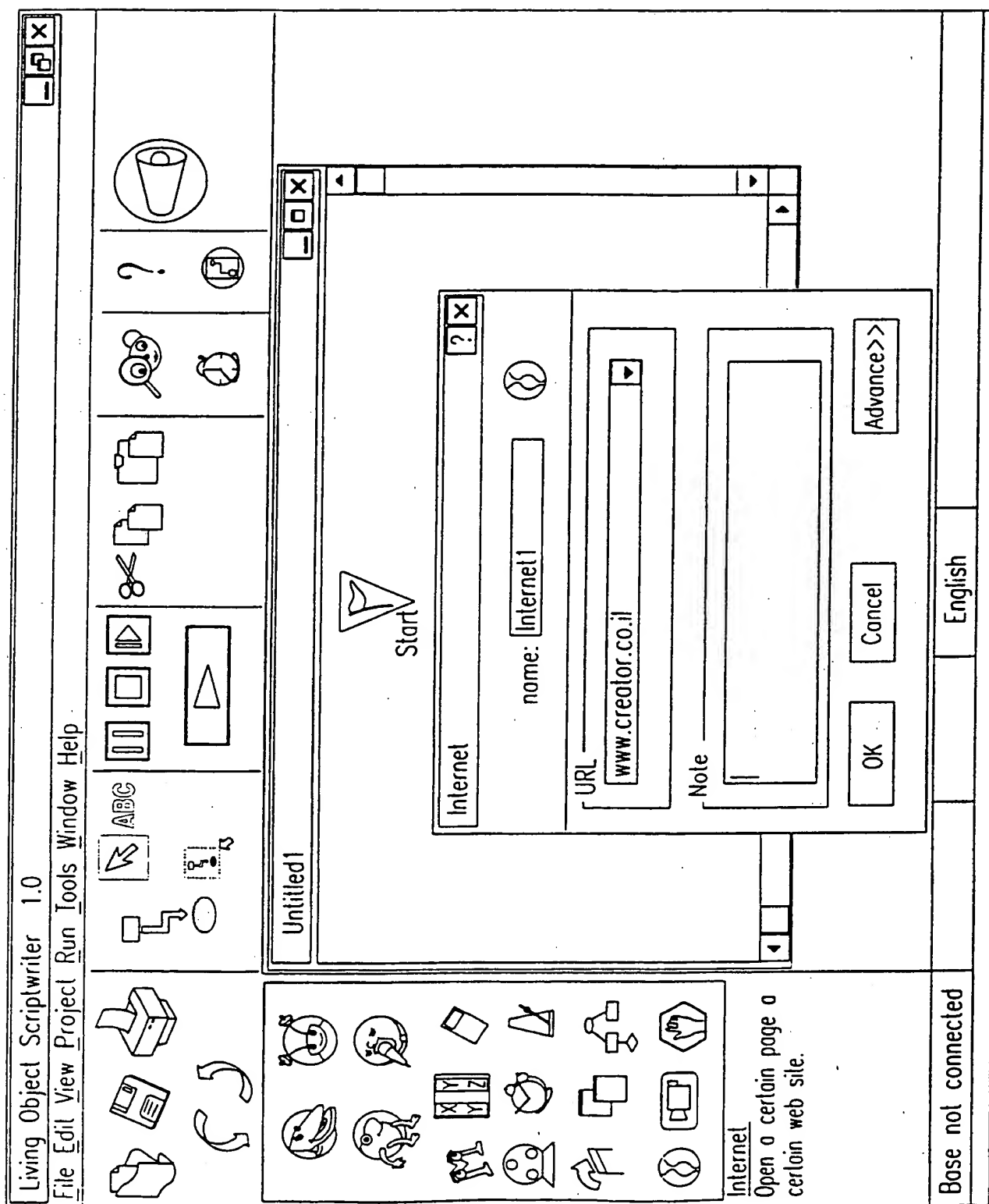
84/120

FIG. 89



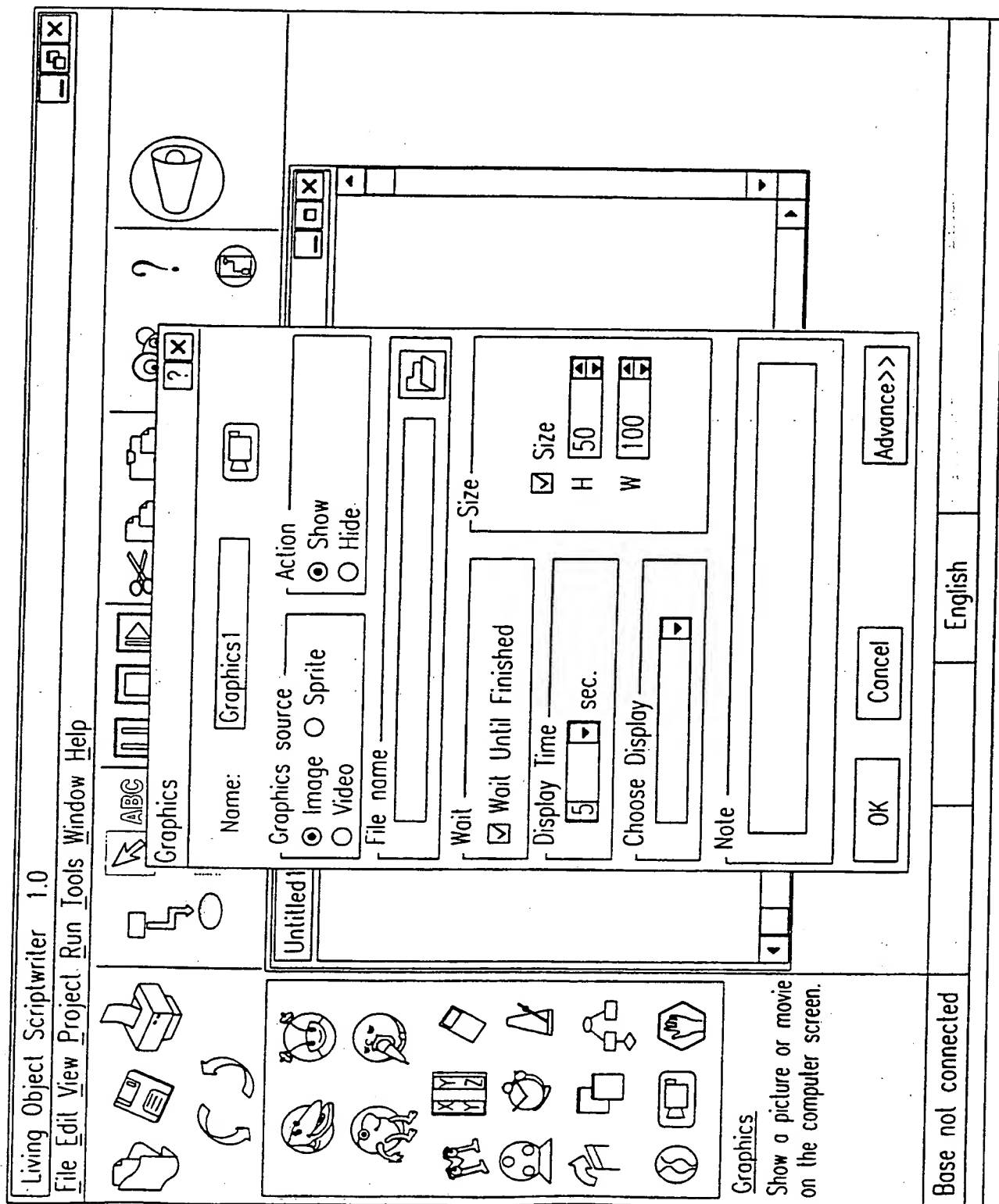
85/120

FIG. 90



86/120

FIG. 91



87/120

FIG. 92

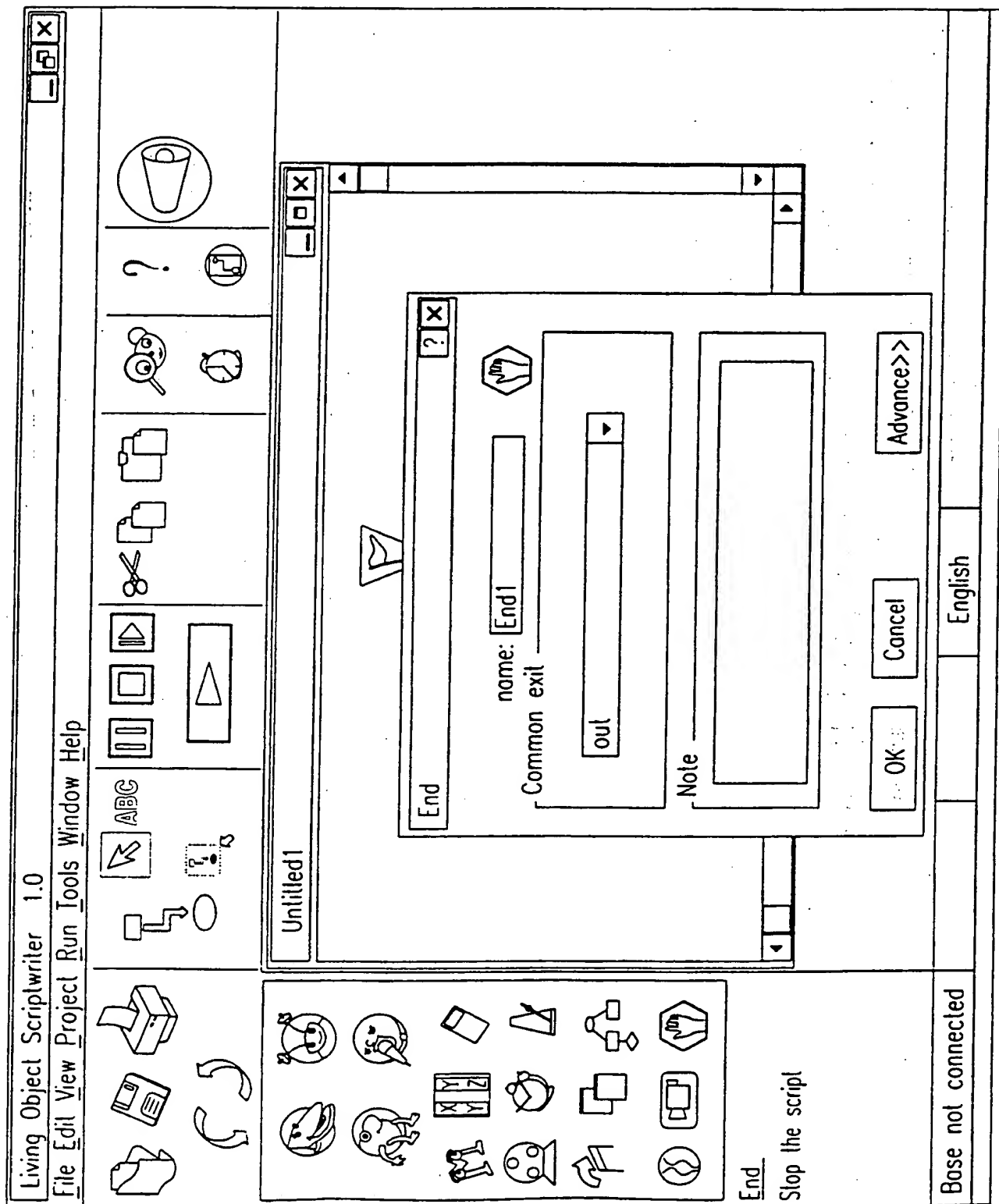
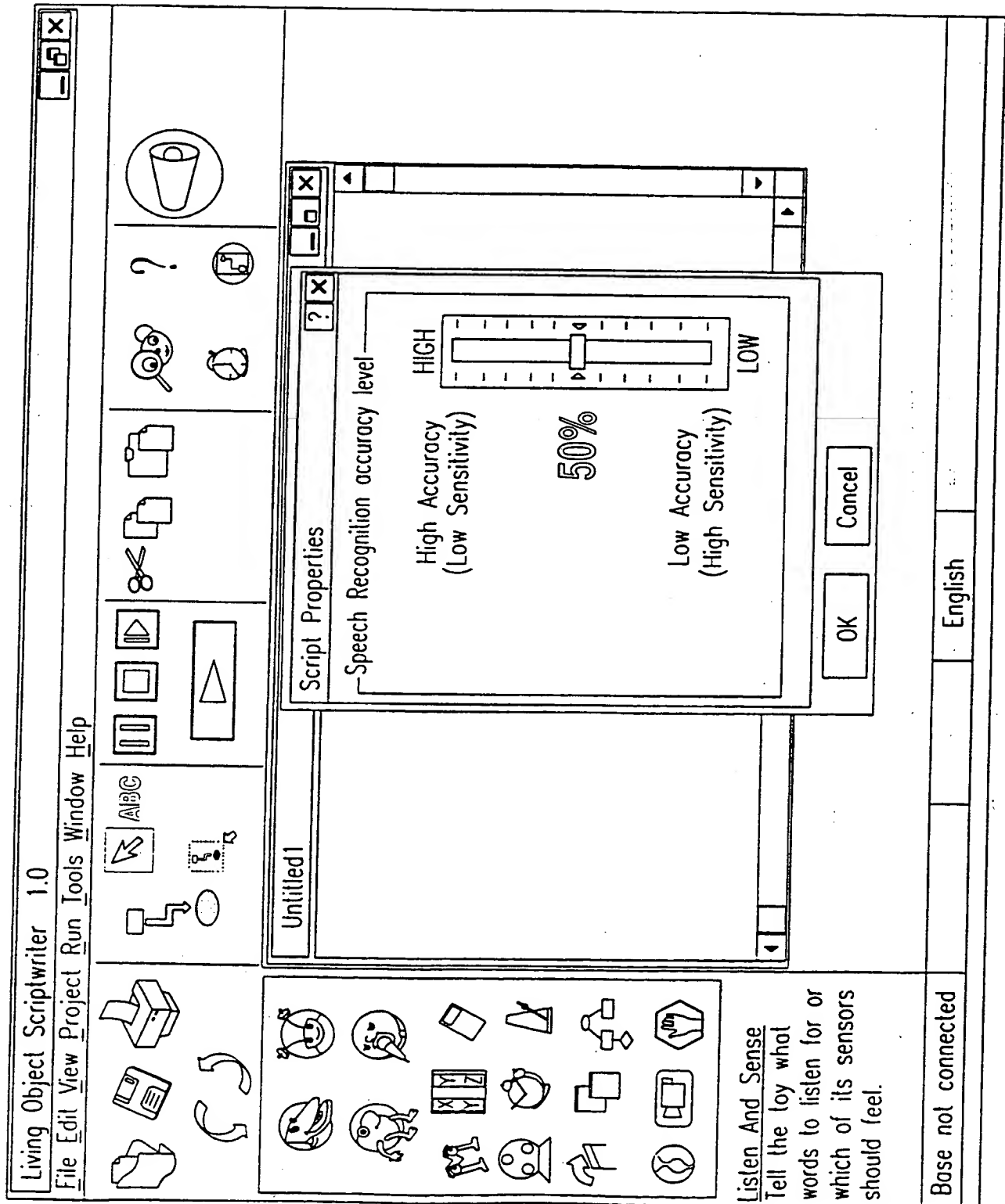
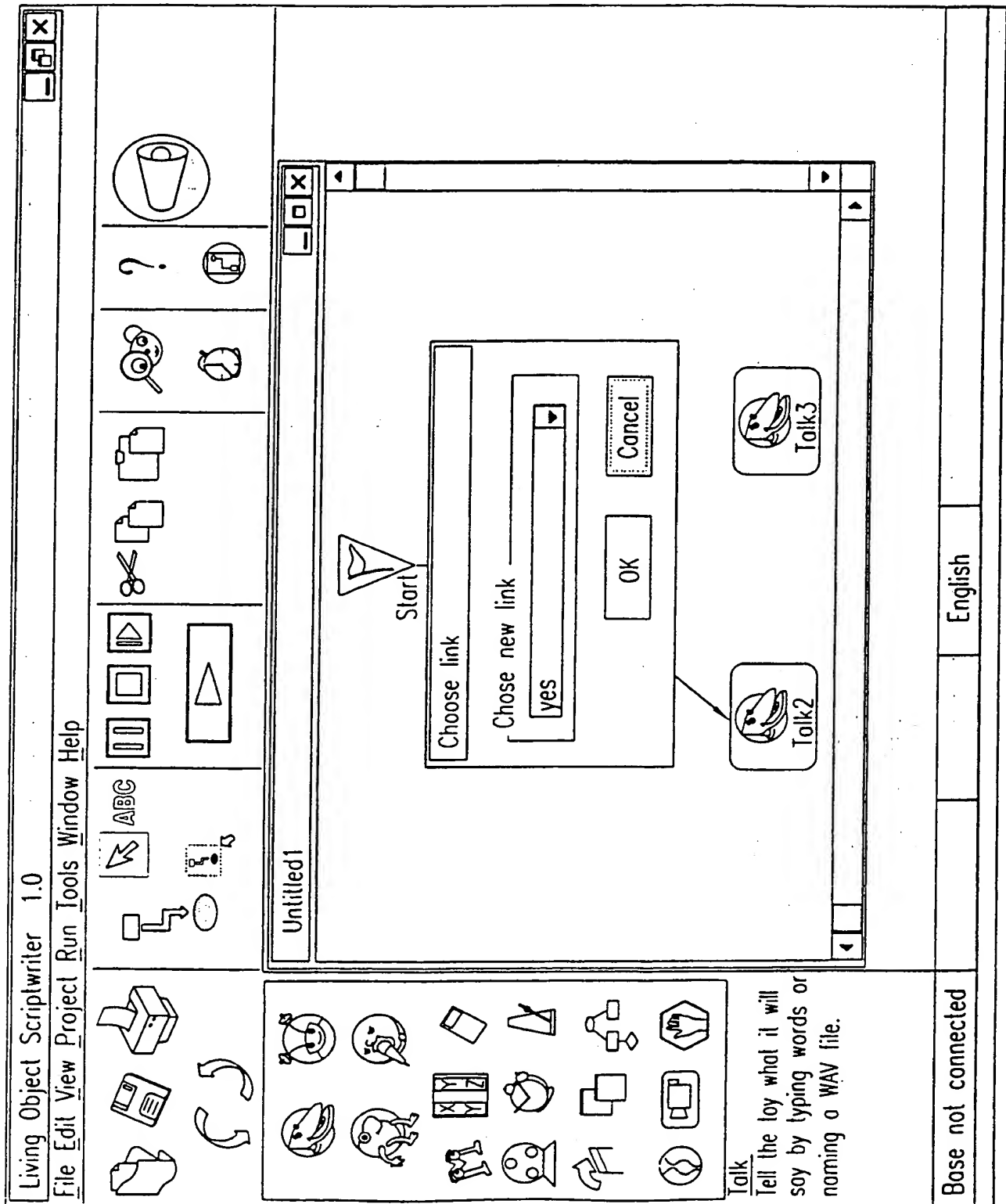


FIG. 93



89/120

FIG. 94



90/120

FIG. 95

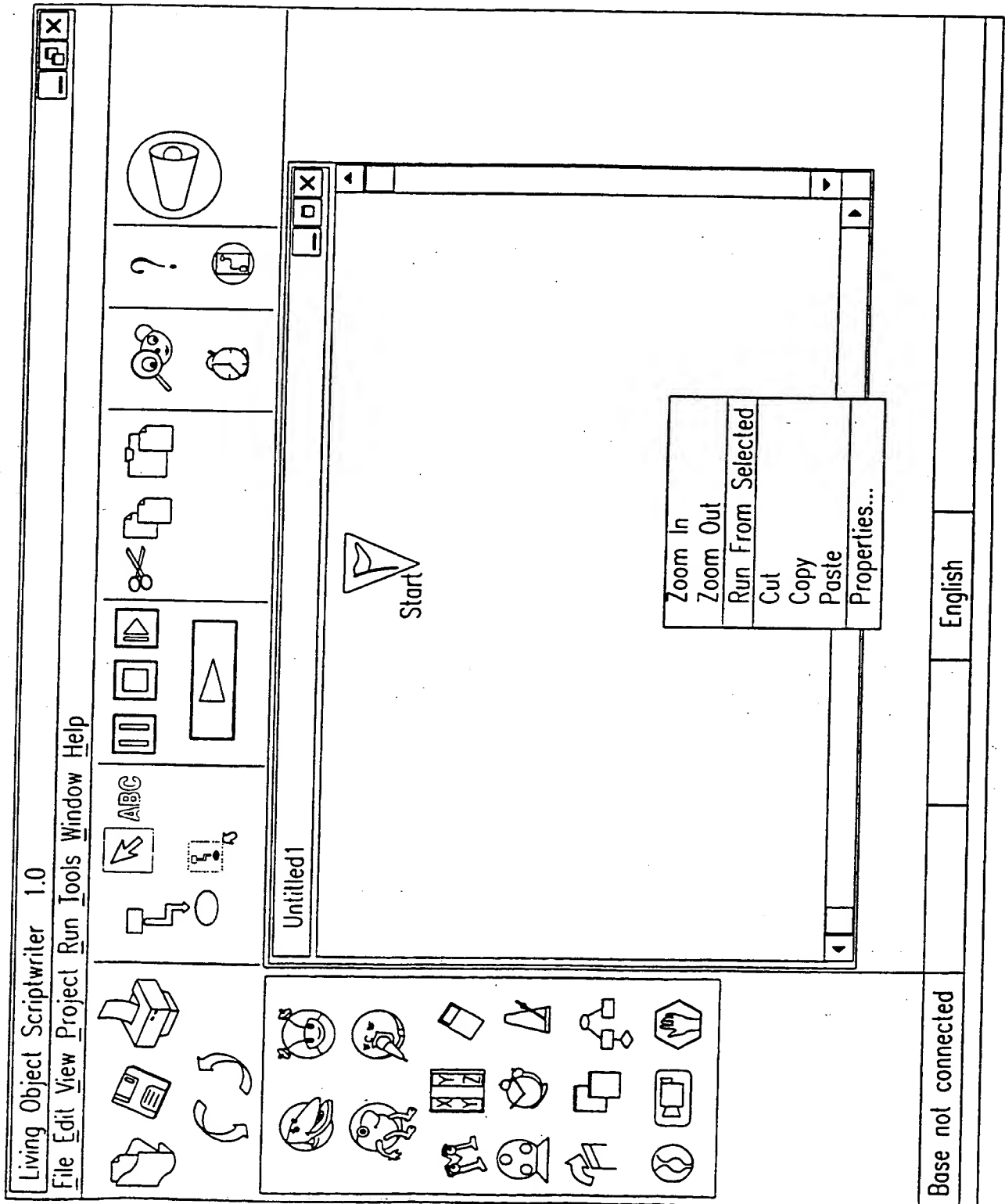


FIG. 96

91/120

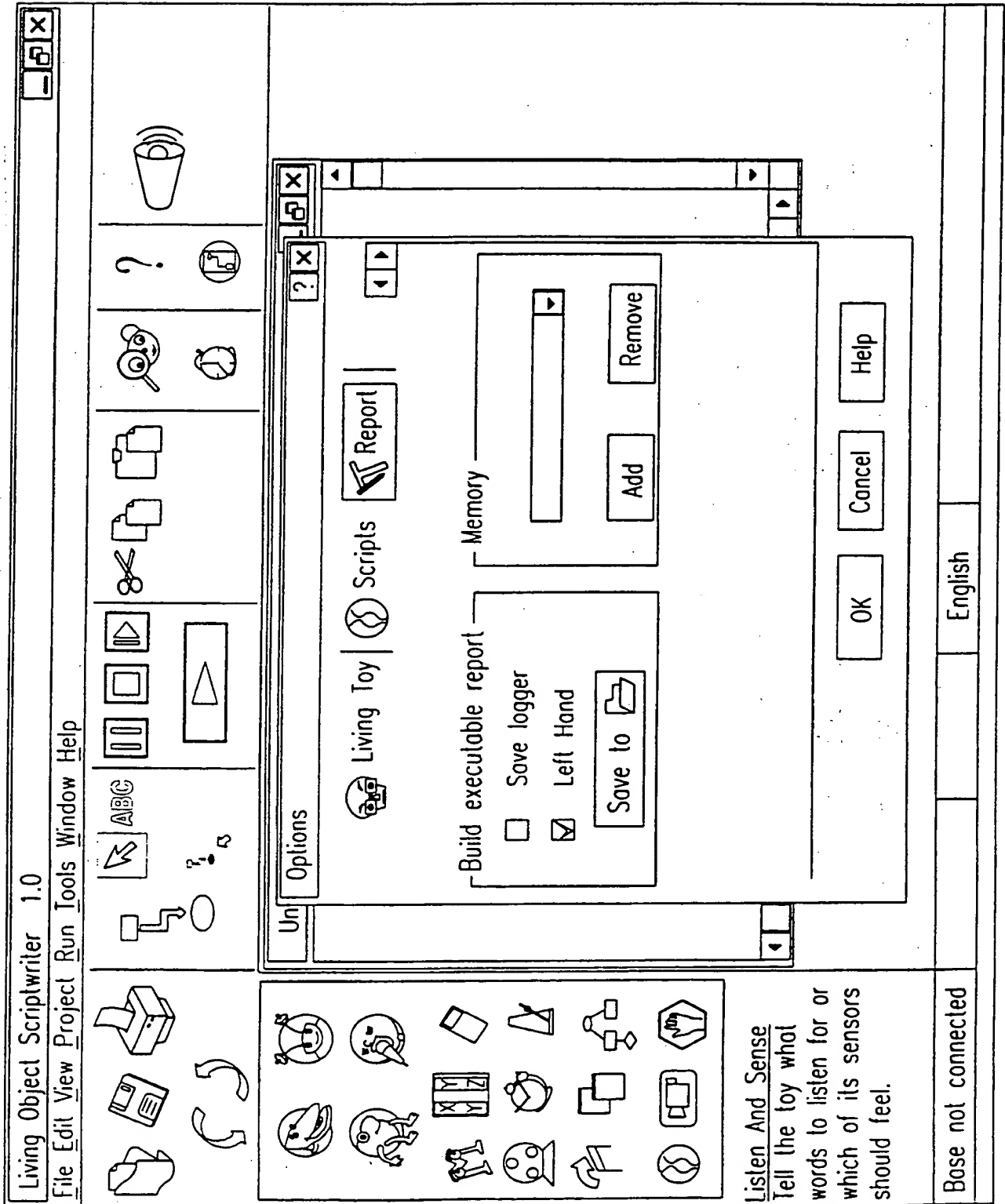
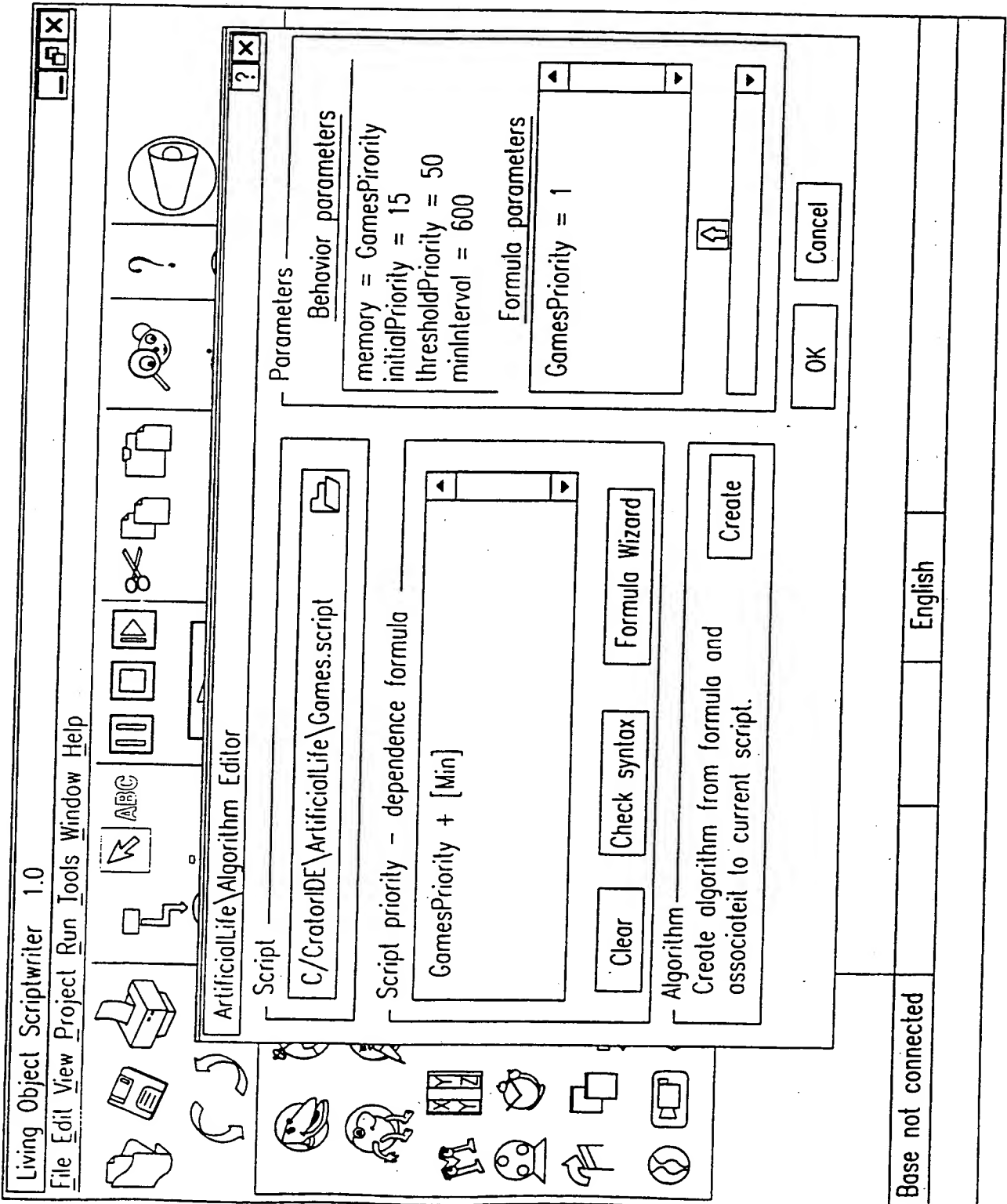


FIG. 97



93/120

FIG. 98

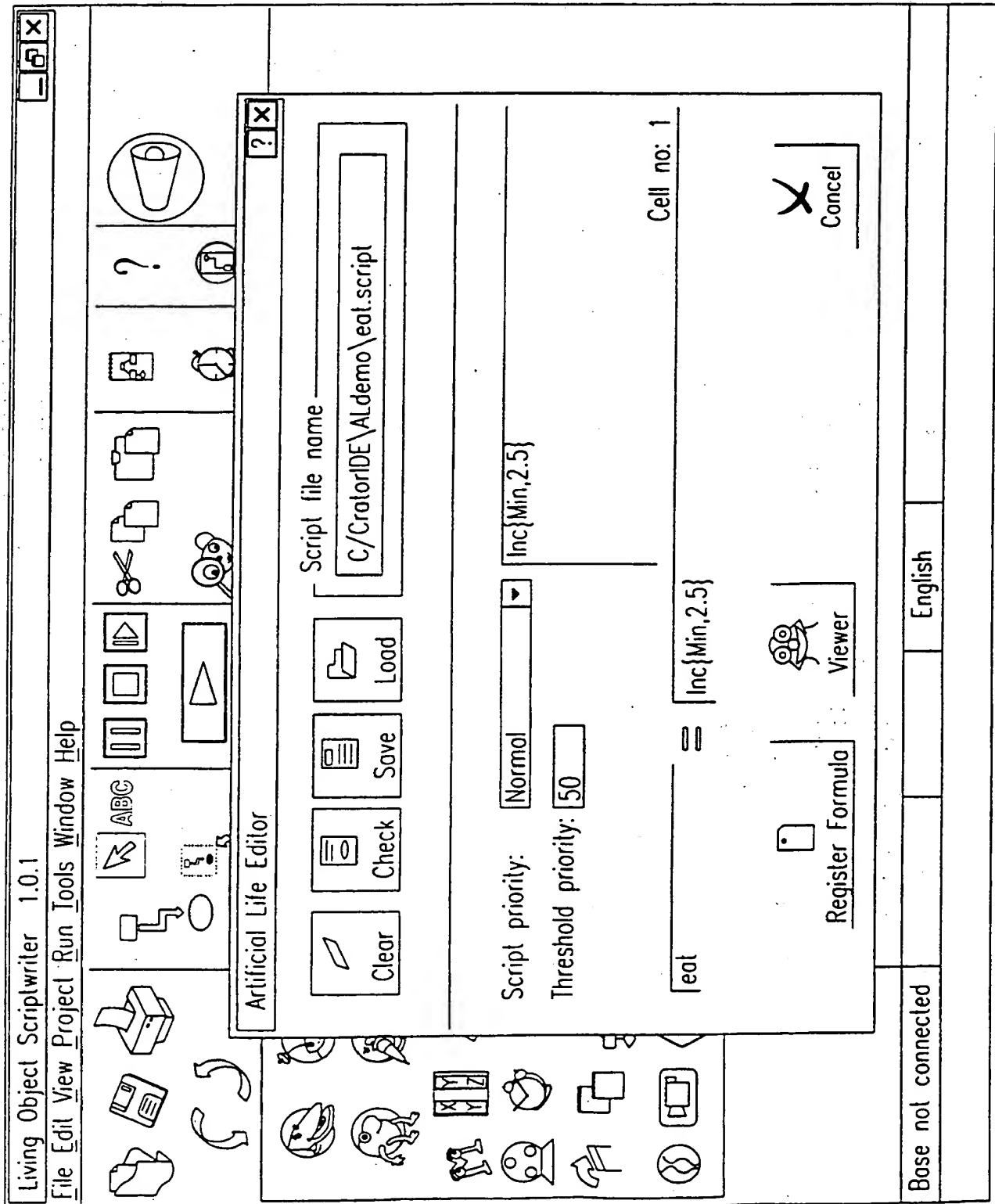
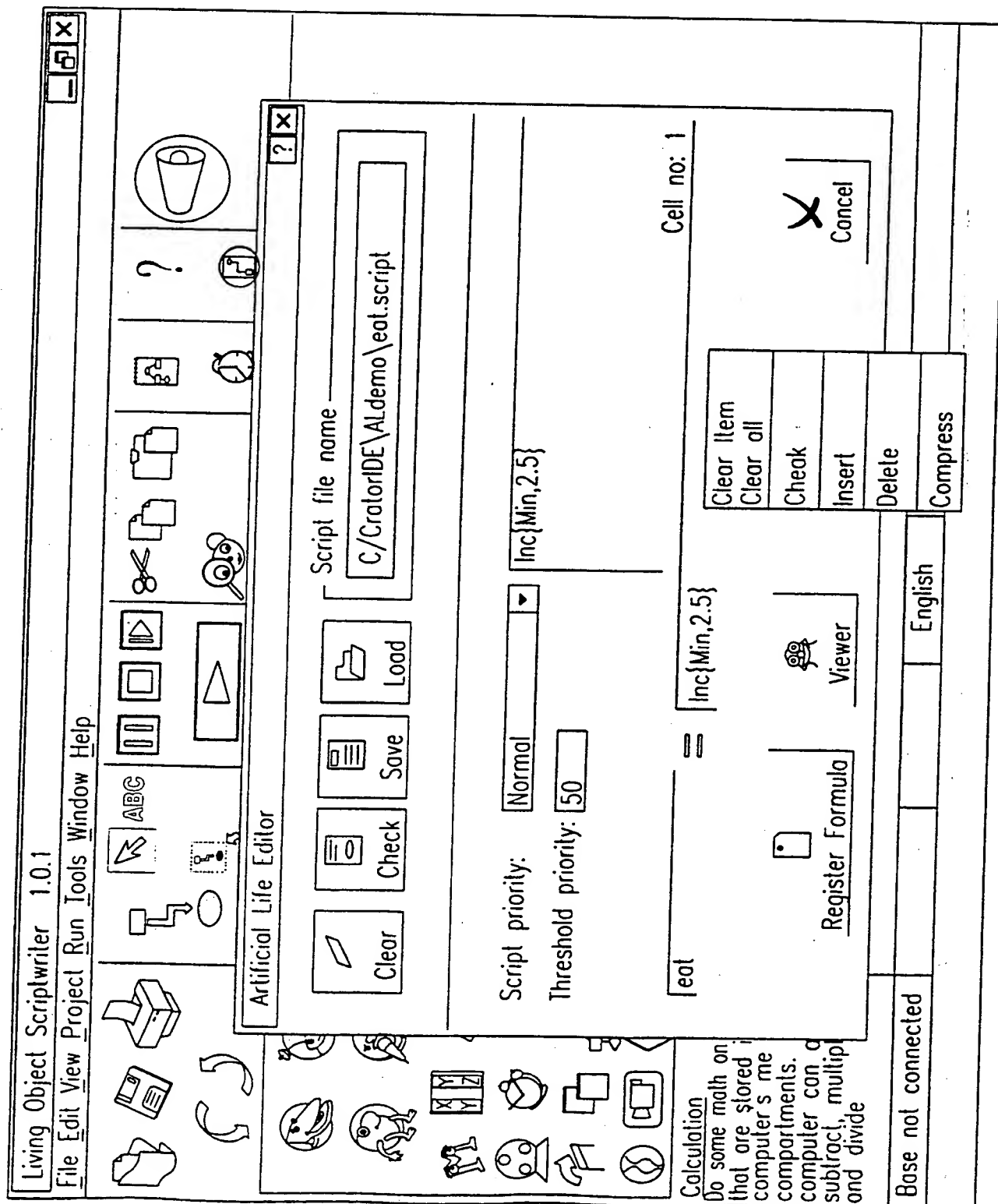
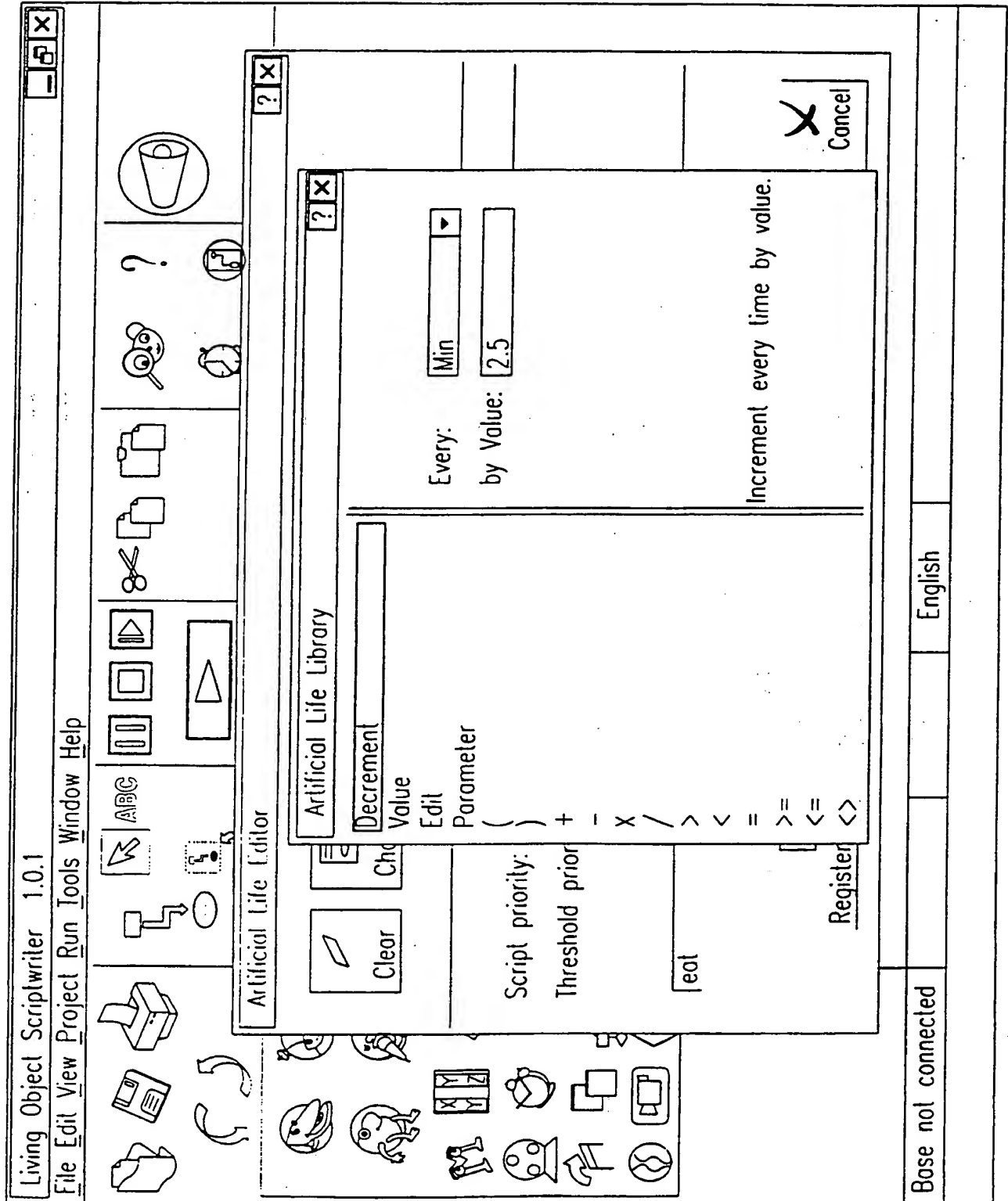


FIG. 99



95/120

FIG. 100



96/120

FIG. 101

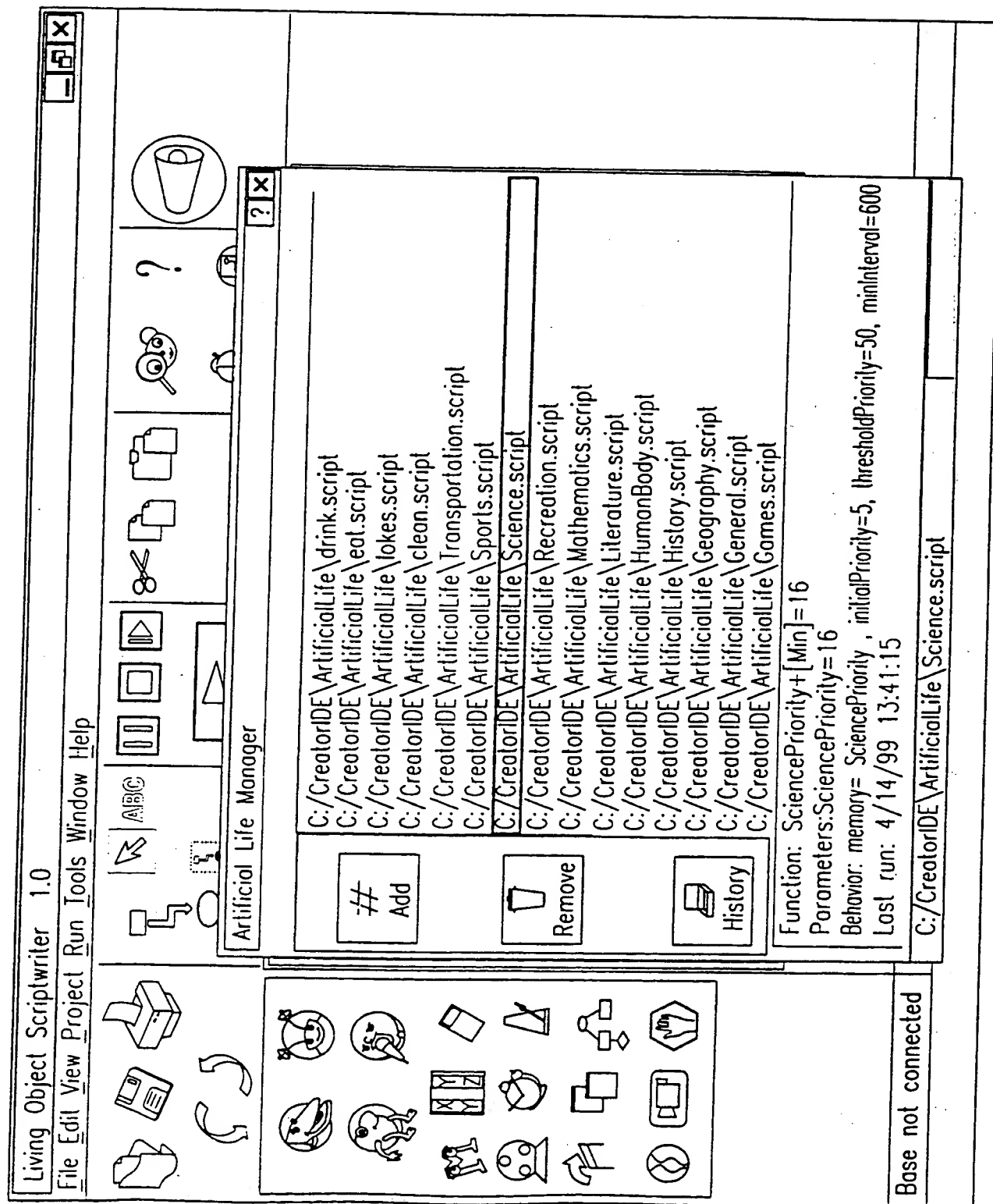


FIG. 102

97/120

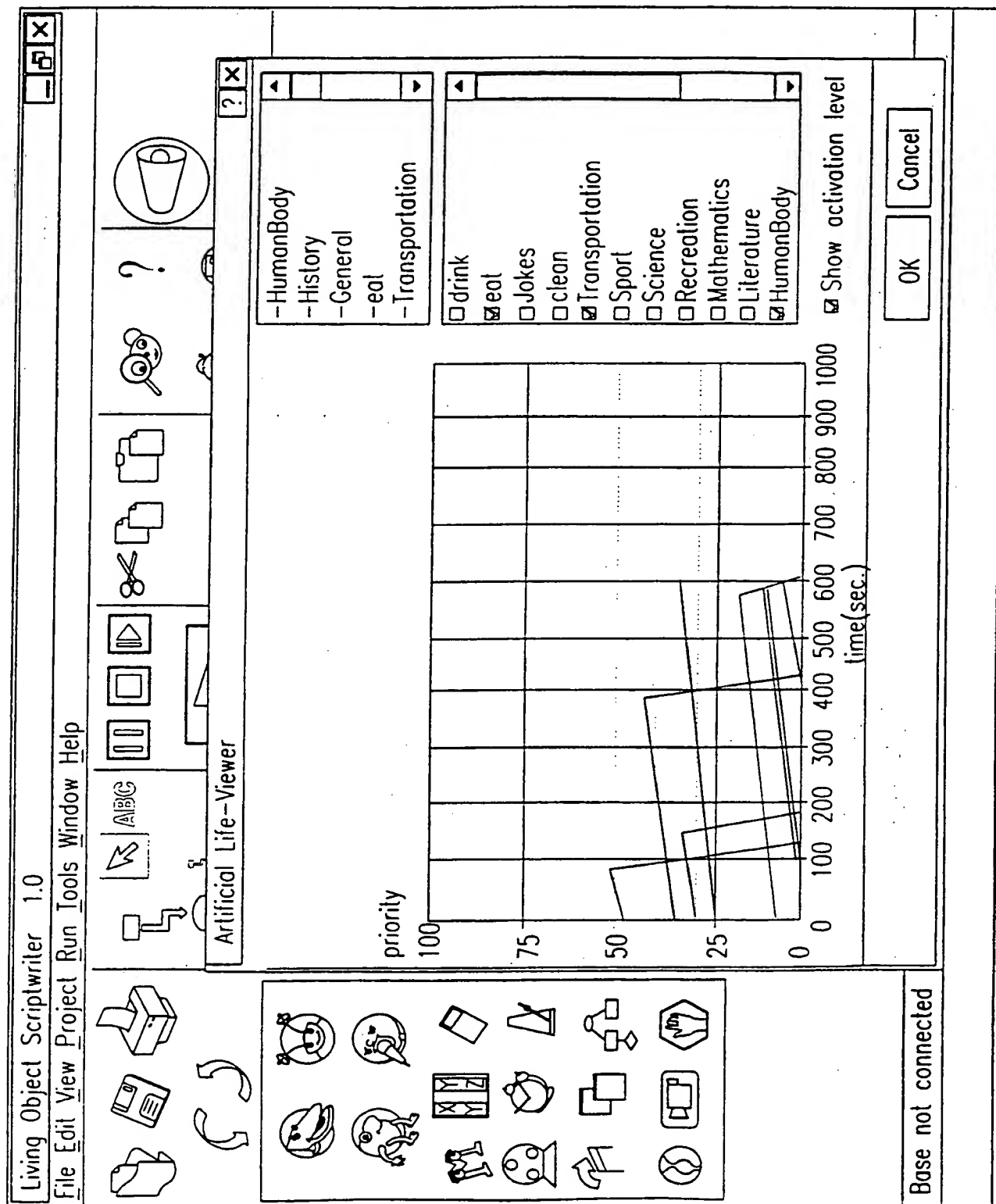


FIG. 103

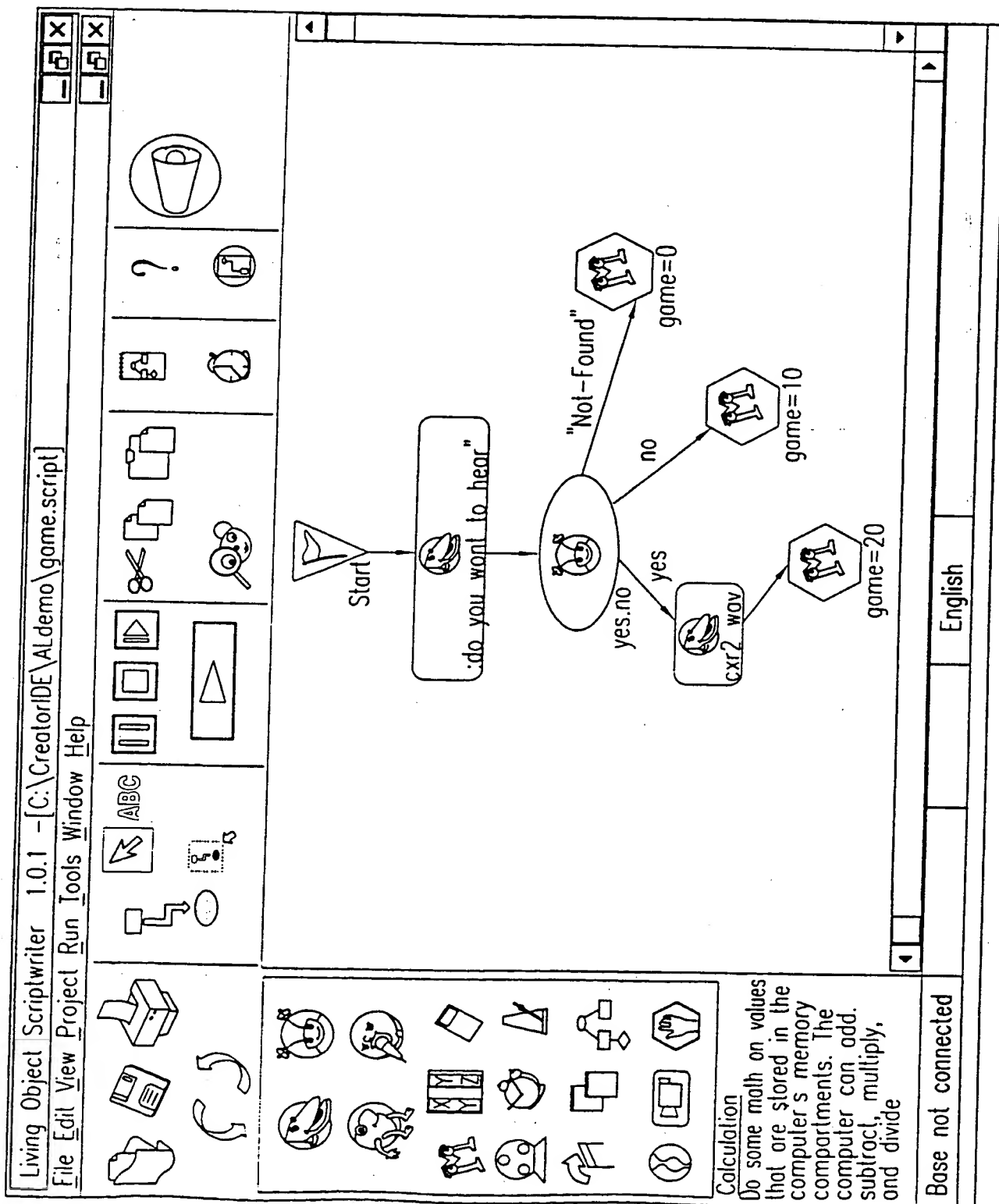


FIG. 104

99/120

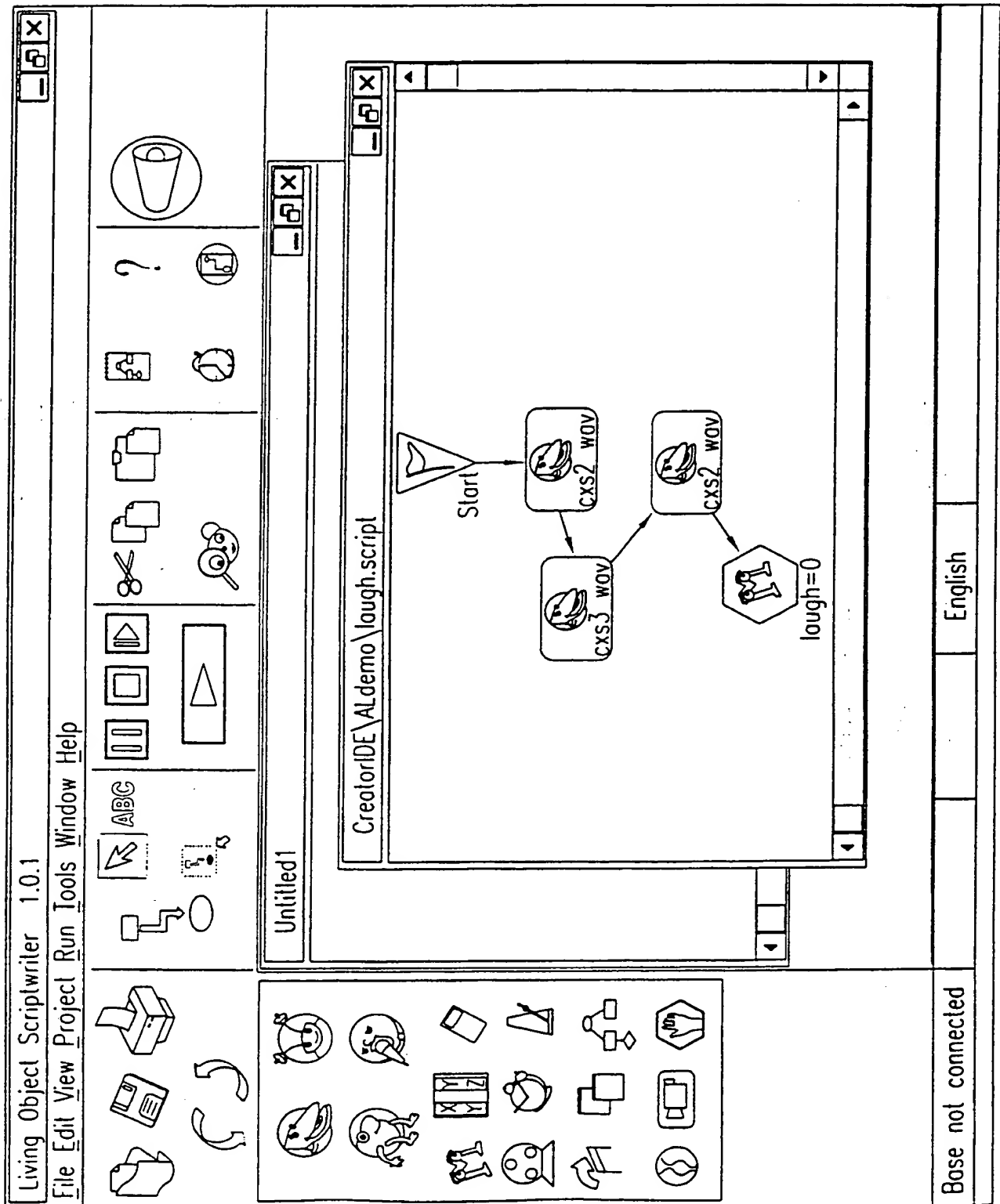


FIG. 105

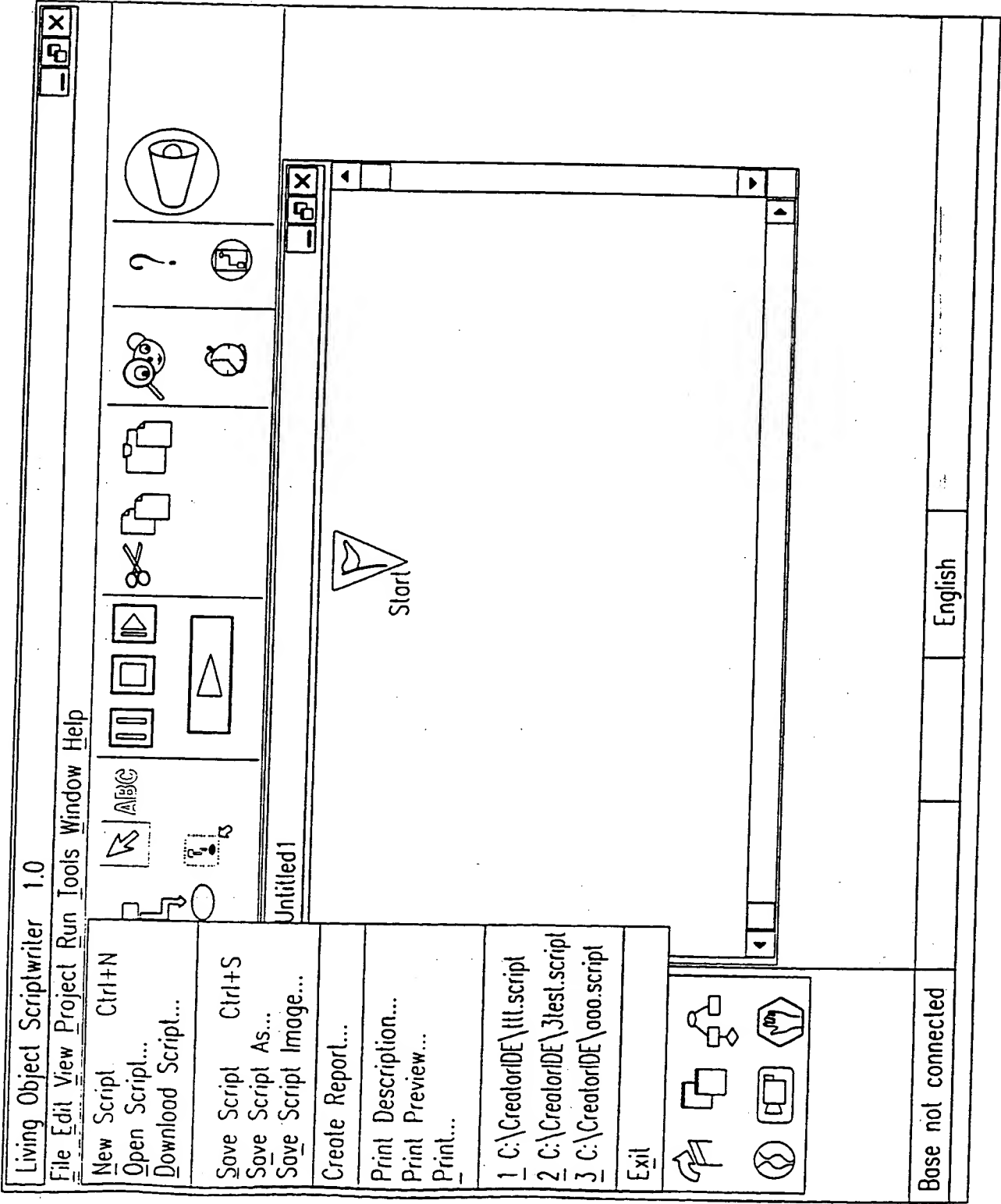


FIG. 106

101/120

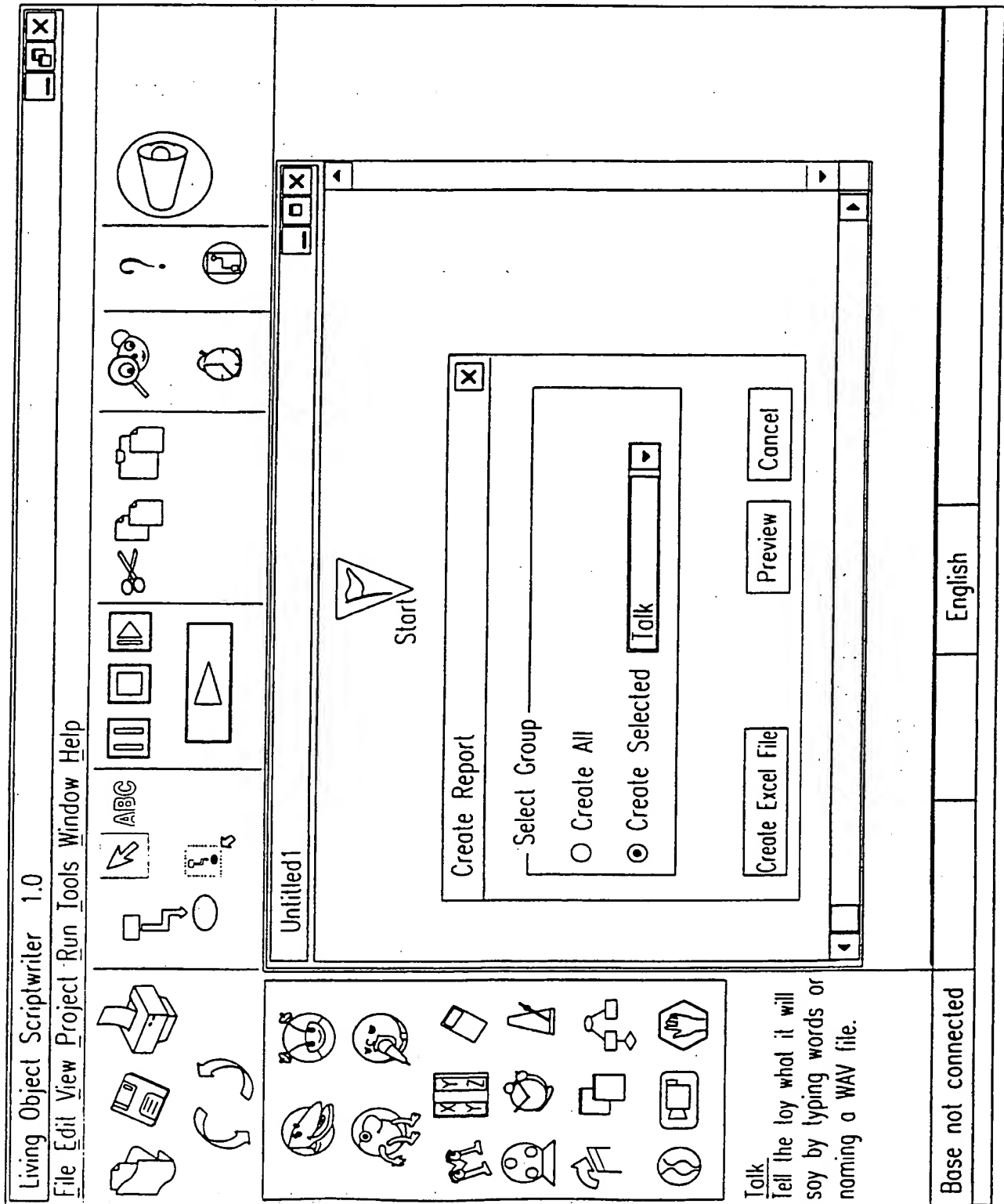
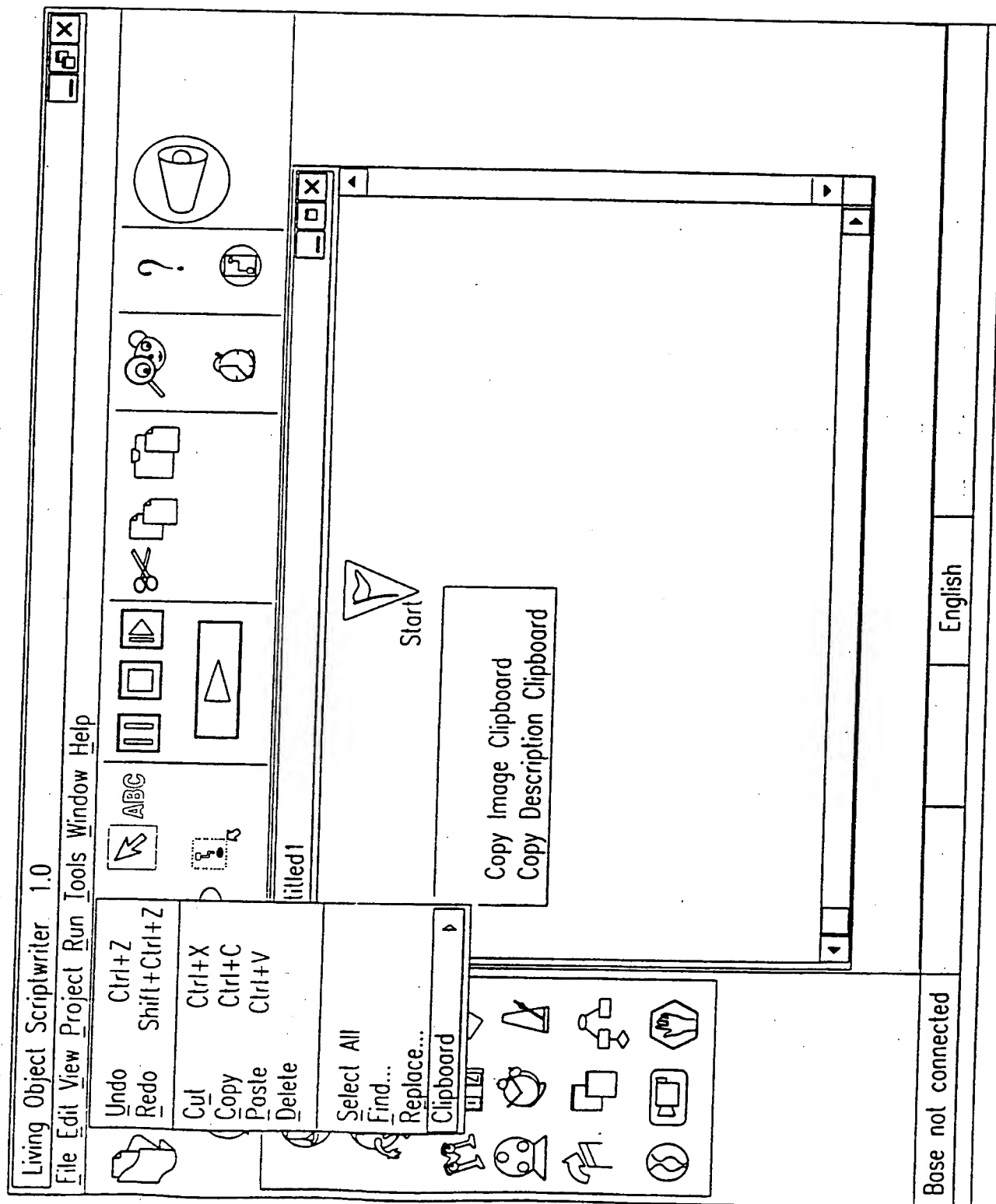


FIG. 107

102/120



103/120

FIG. 108

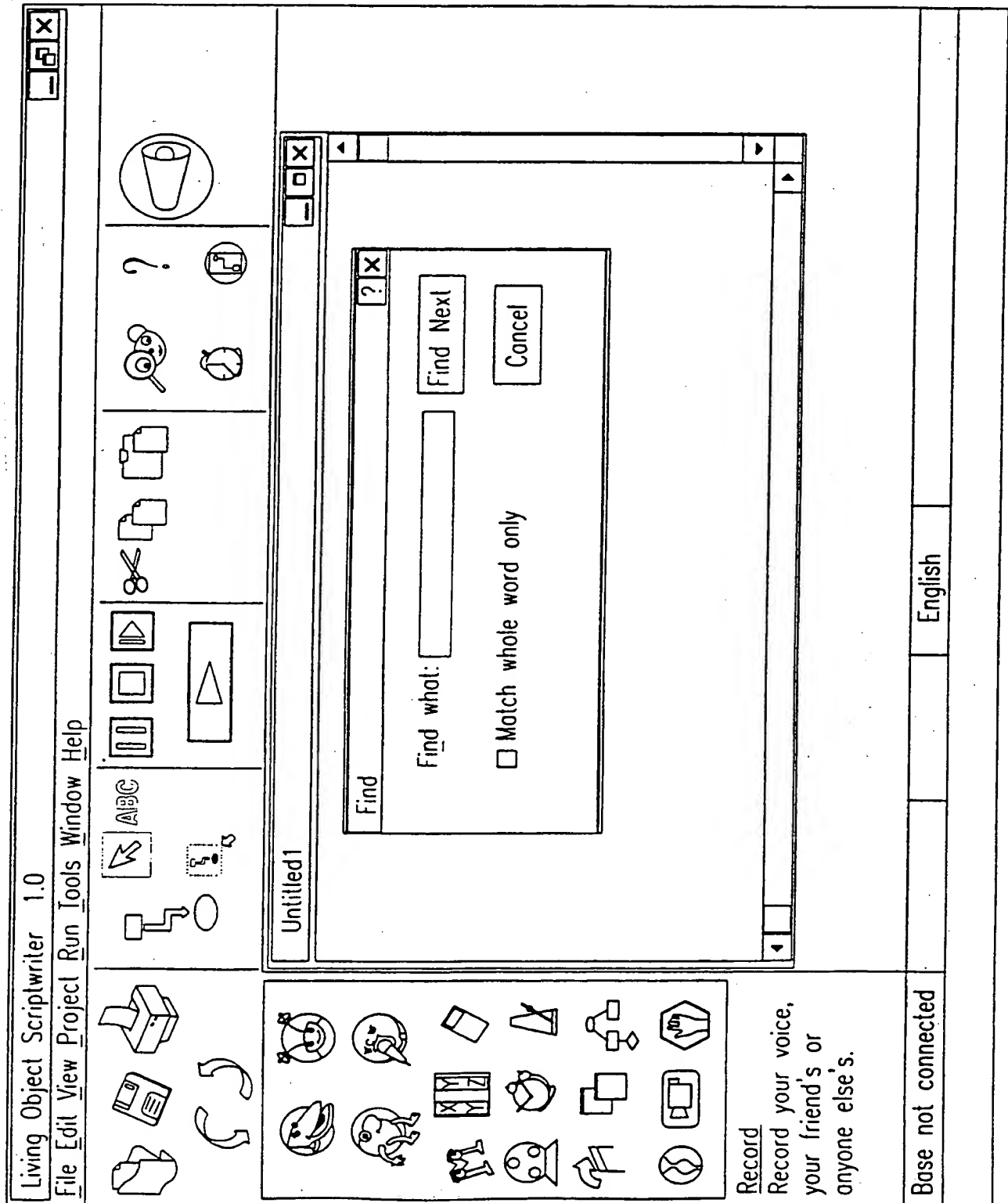


FIG. 109

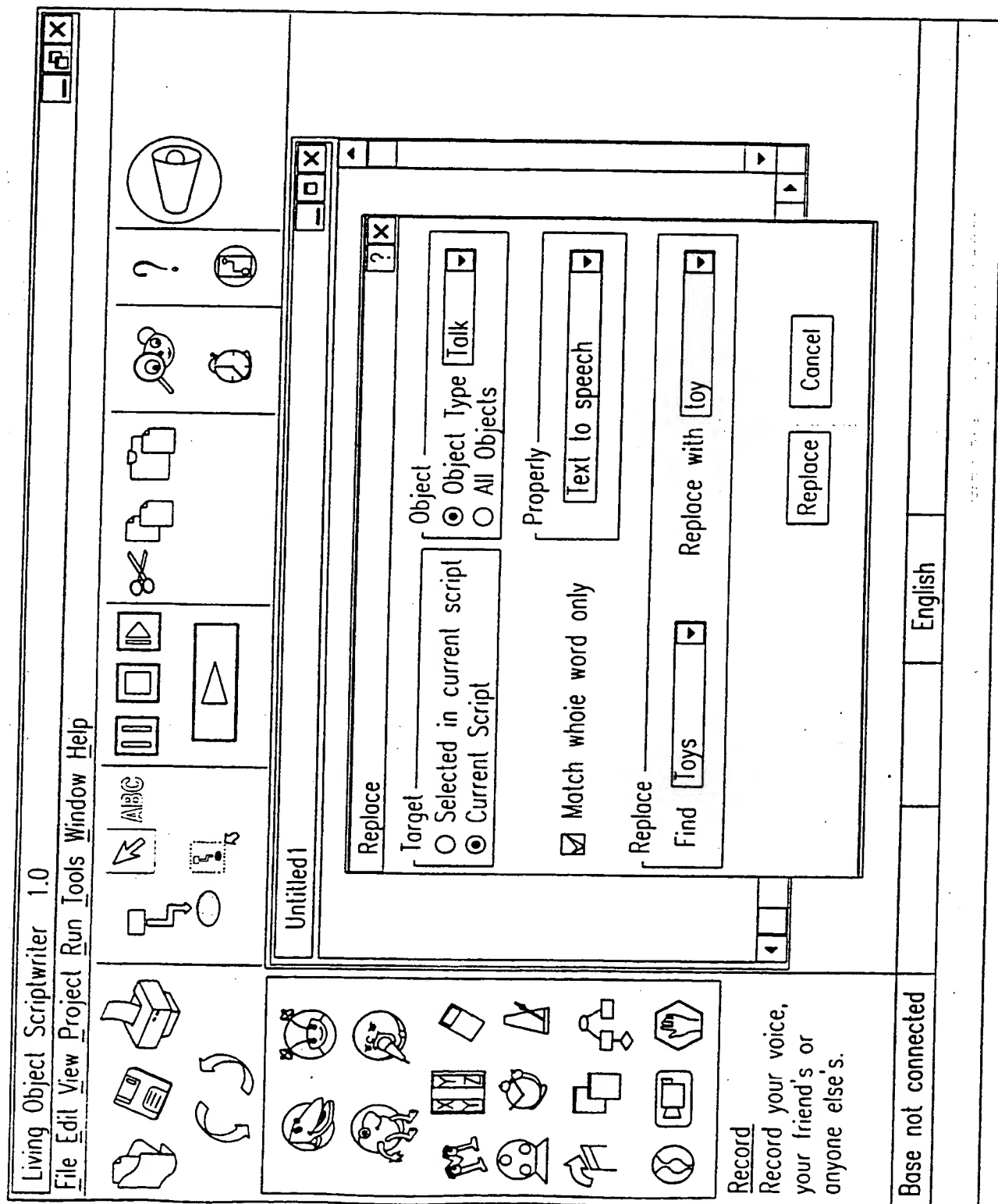


FIG. 110

105/120

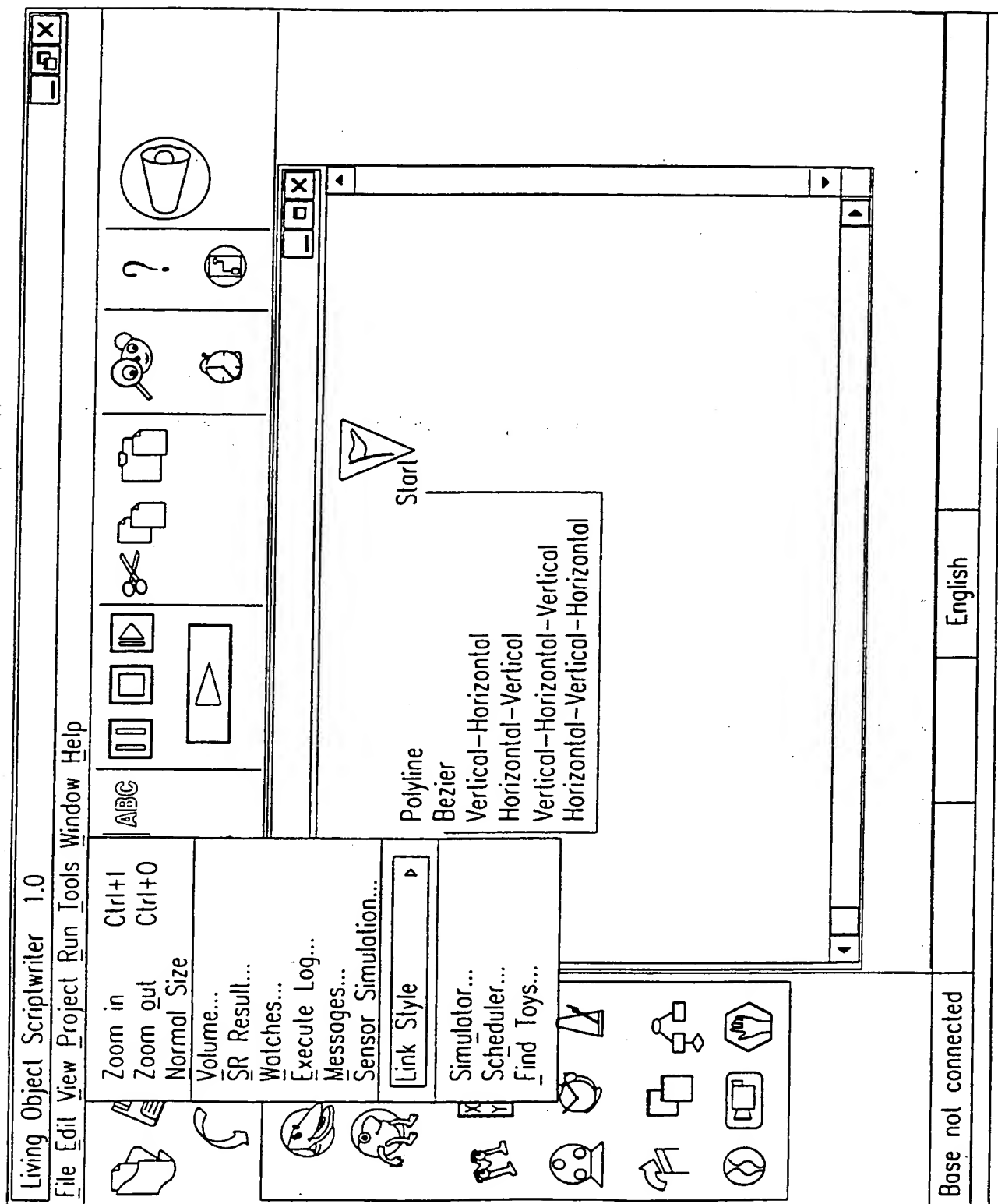
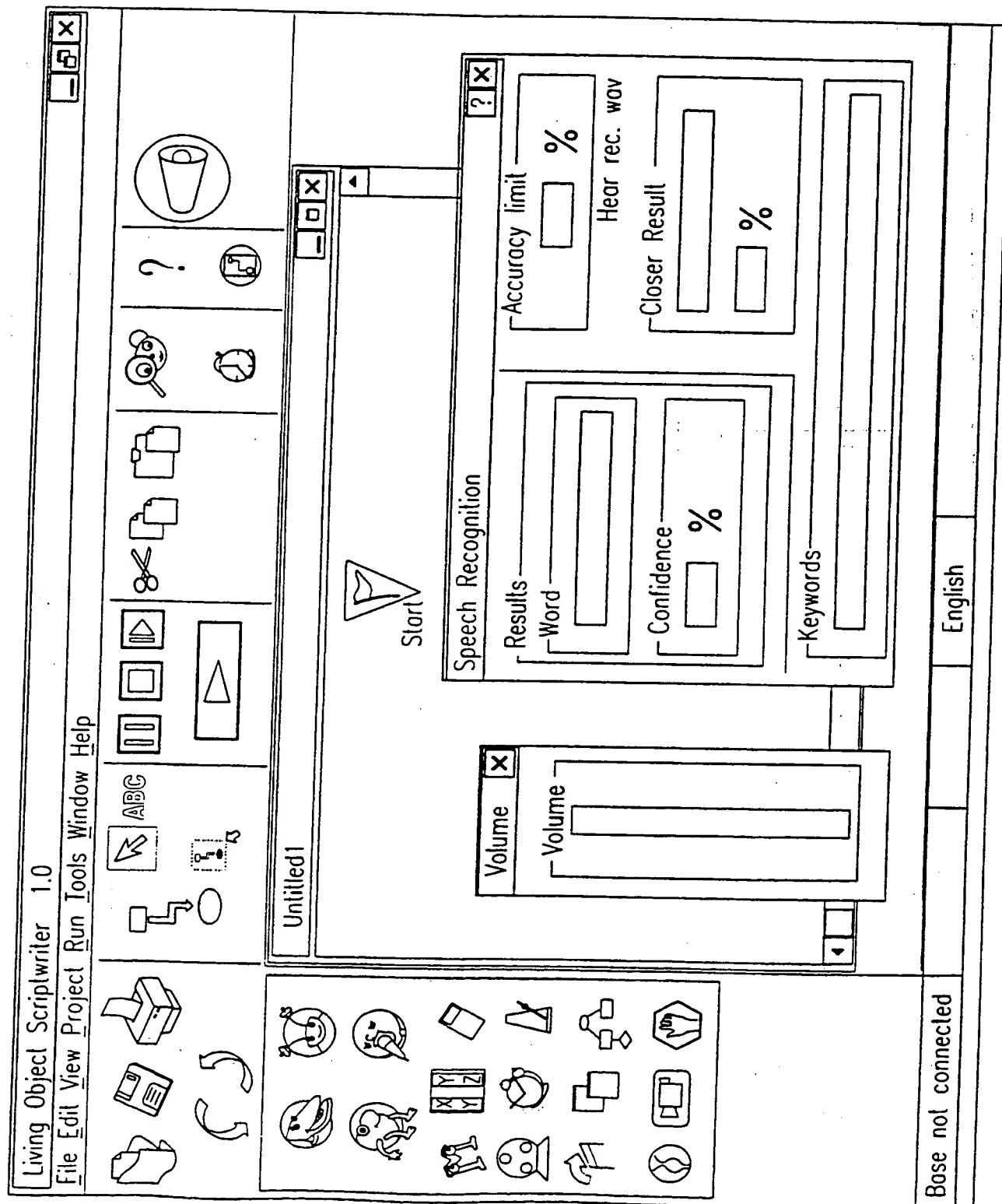


FIG. 111



107/120

FIG. 112

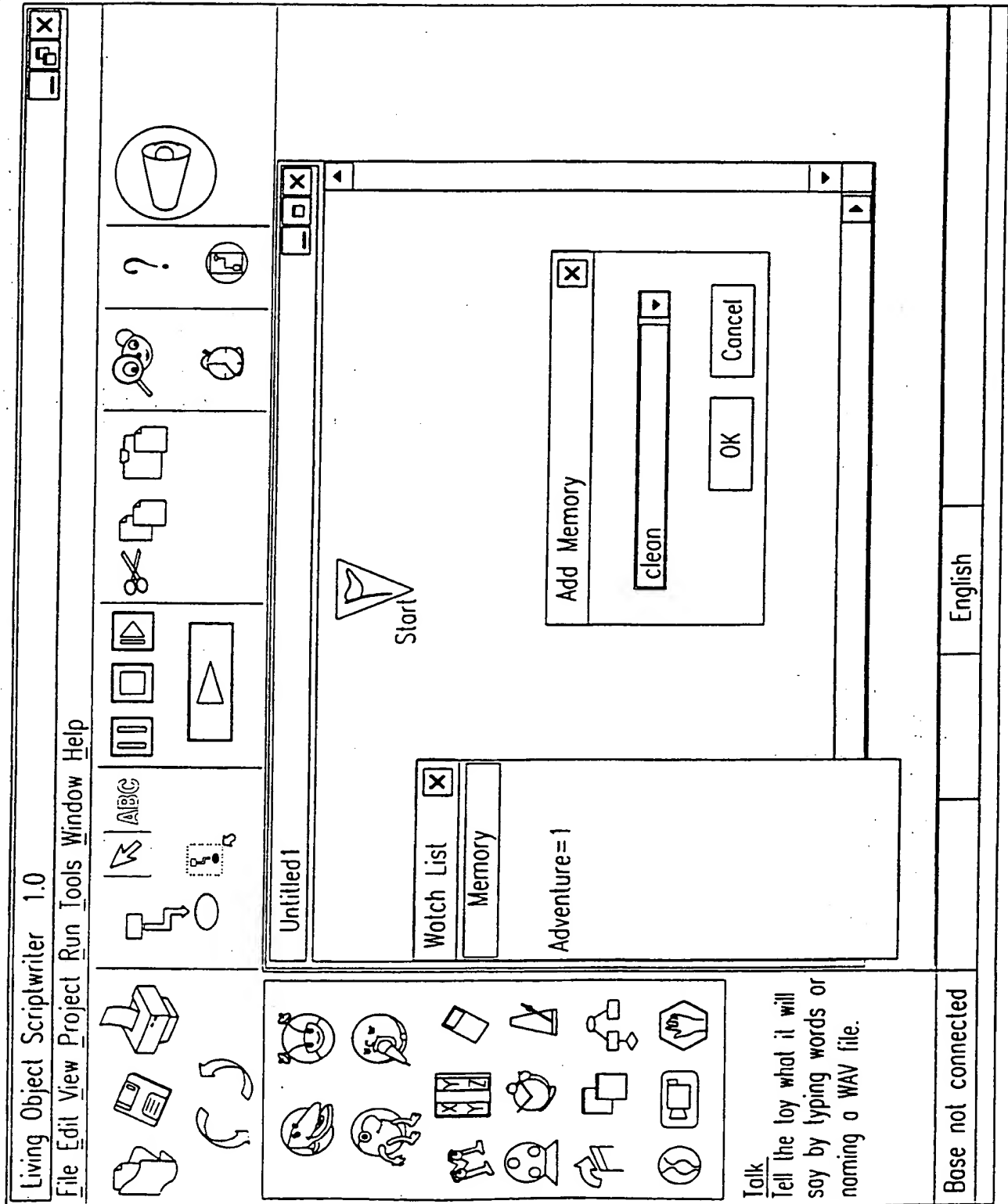


FIG. 113

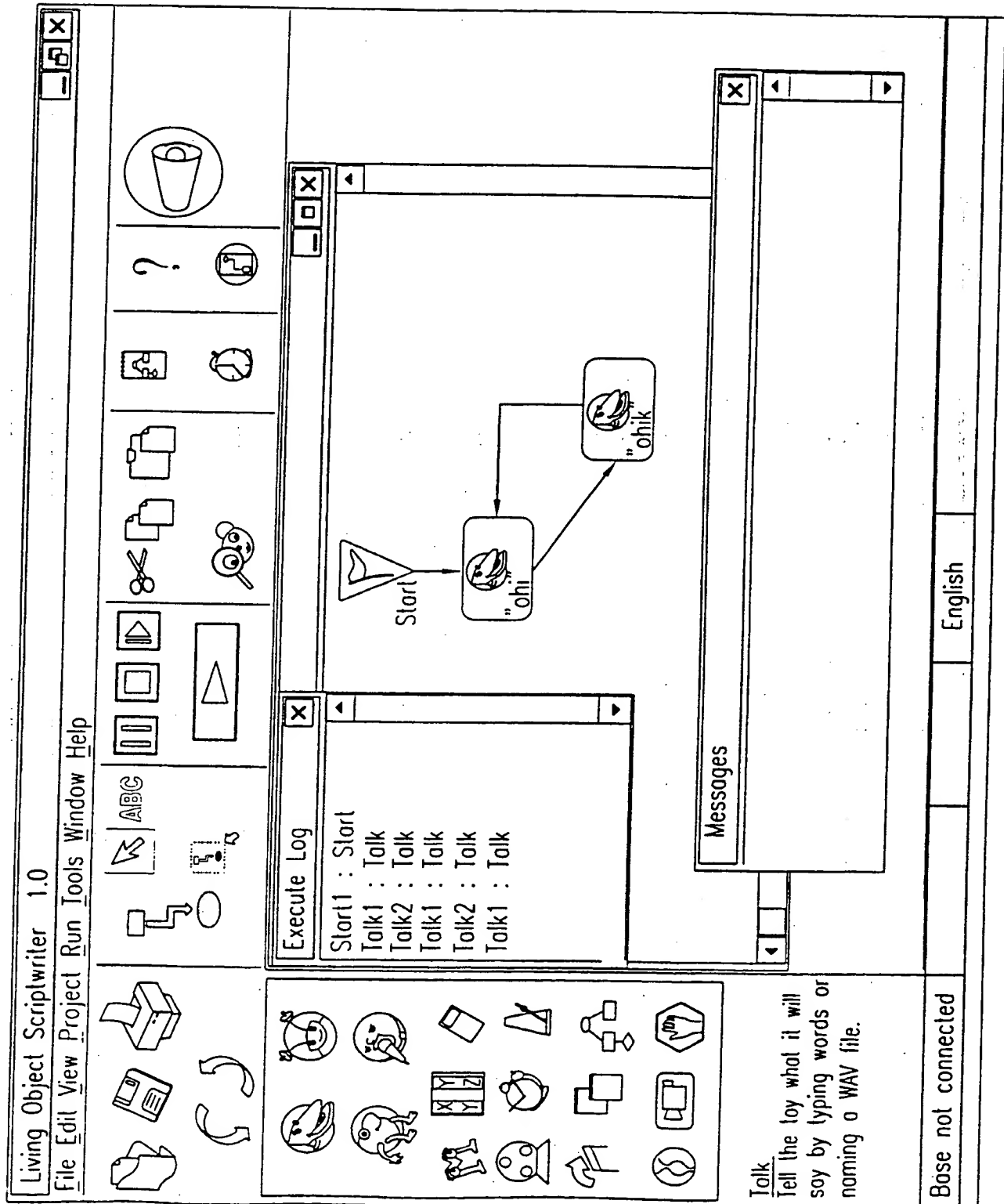
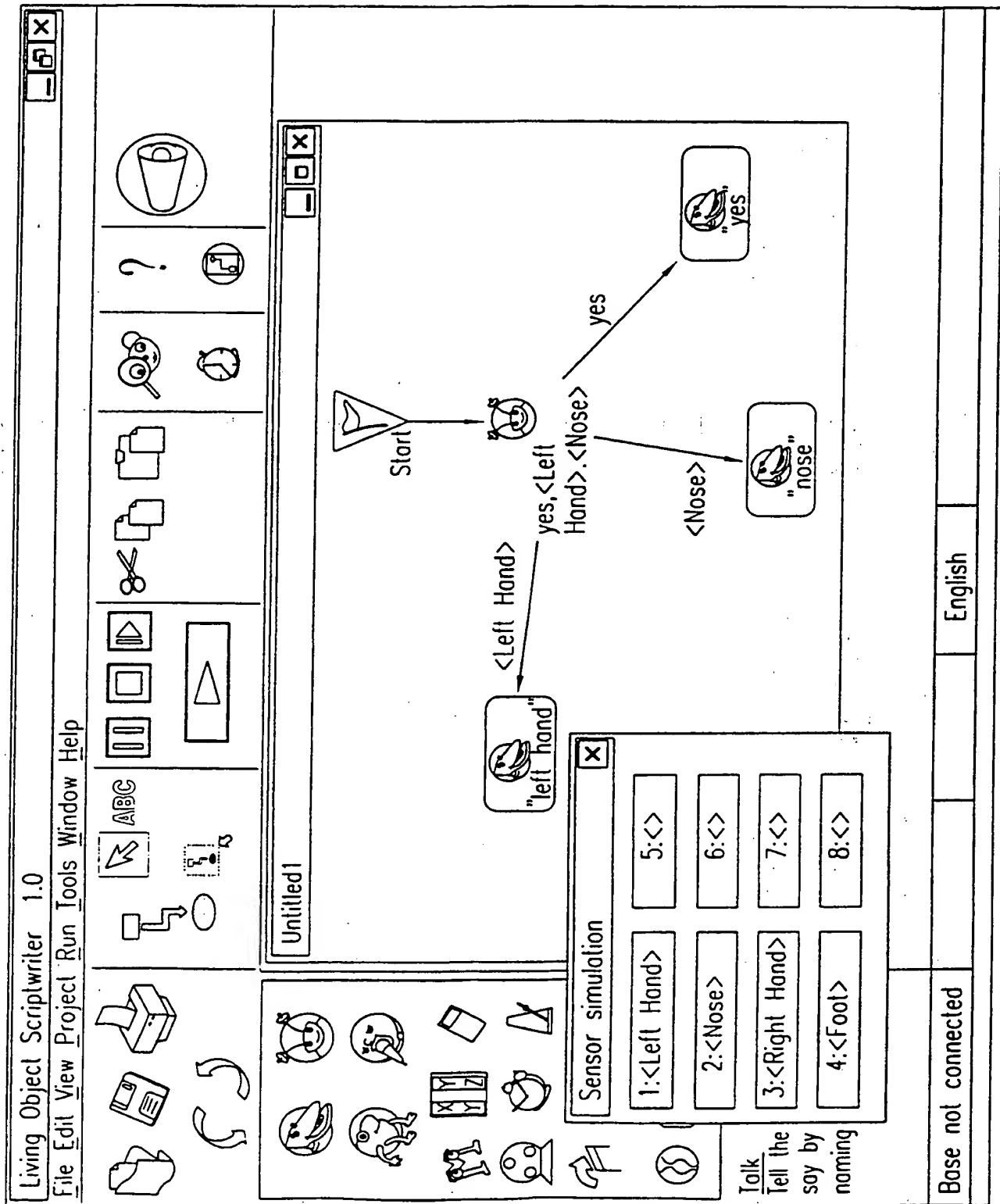


FIG. 114

109/120



110/120

FIG. 115

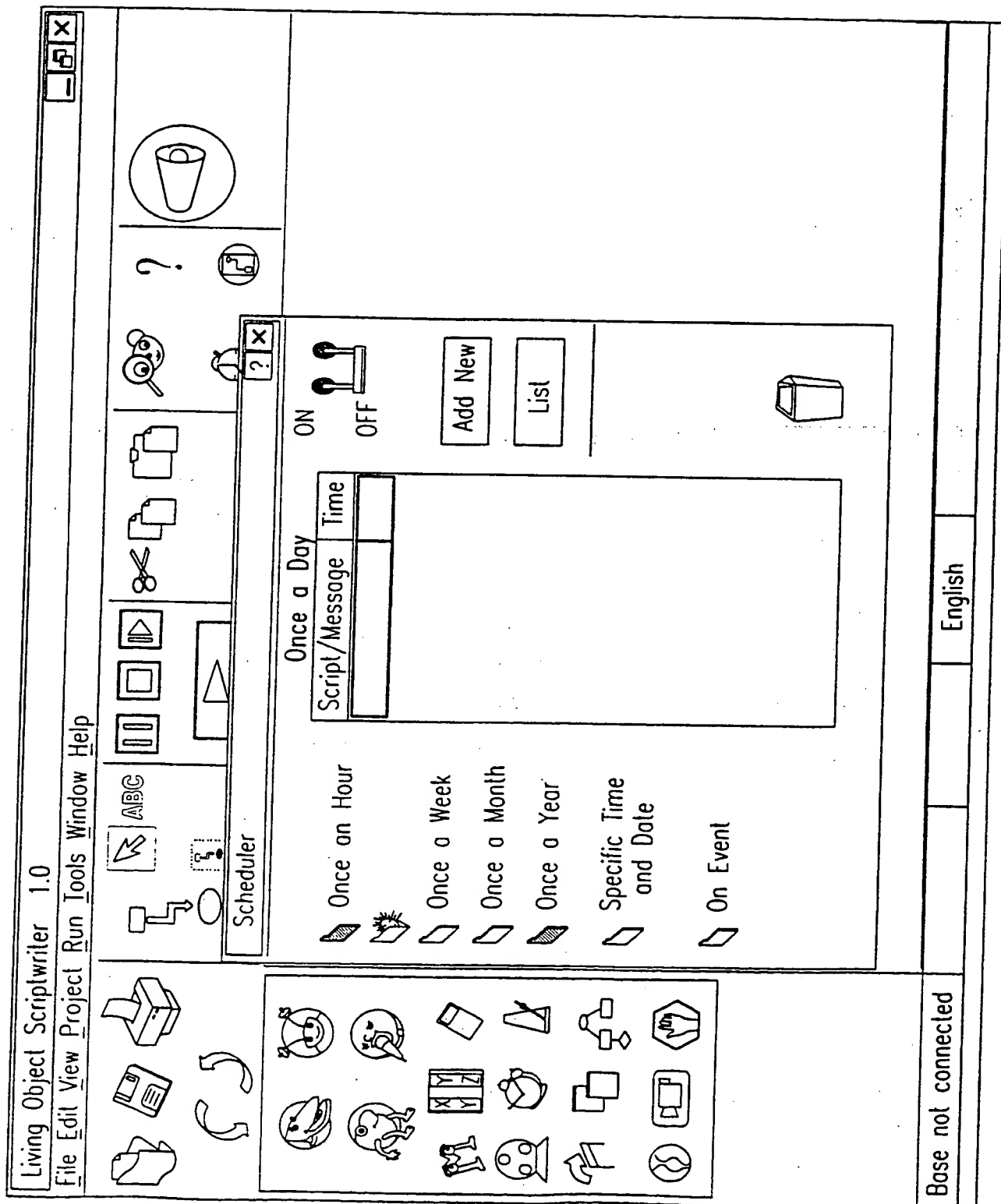


FIG. 116

111/120

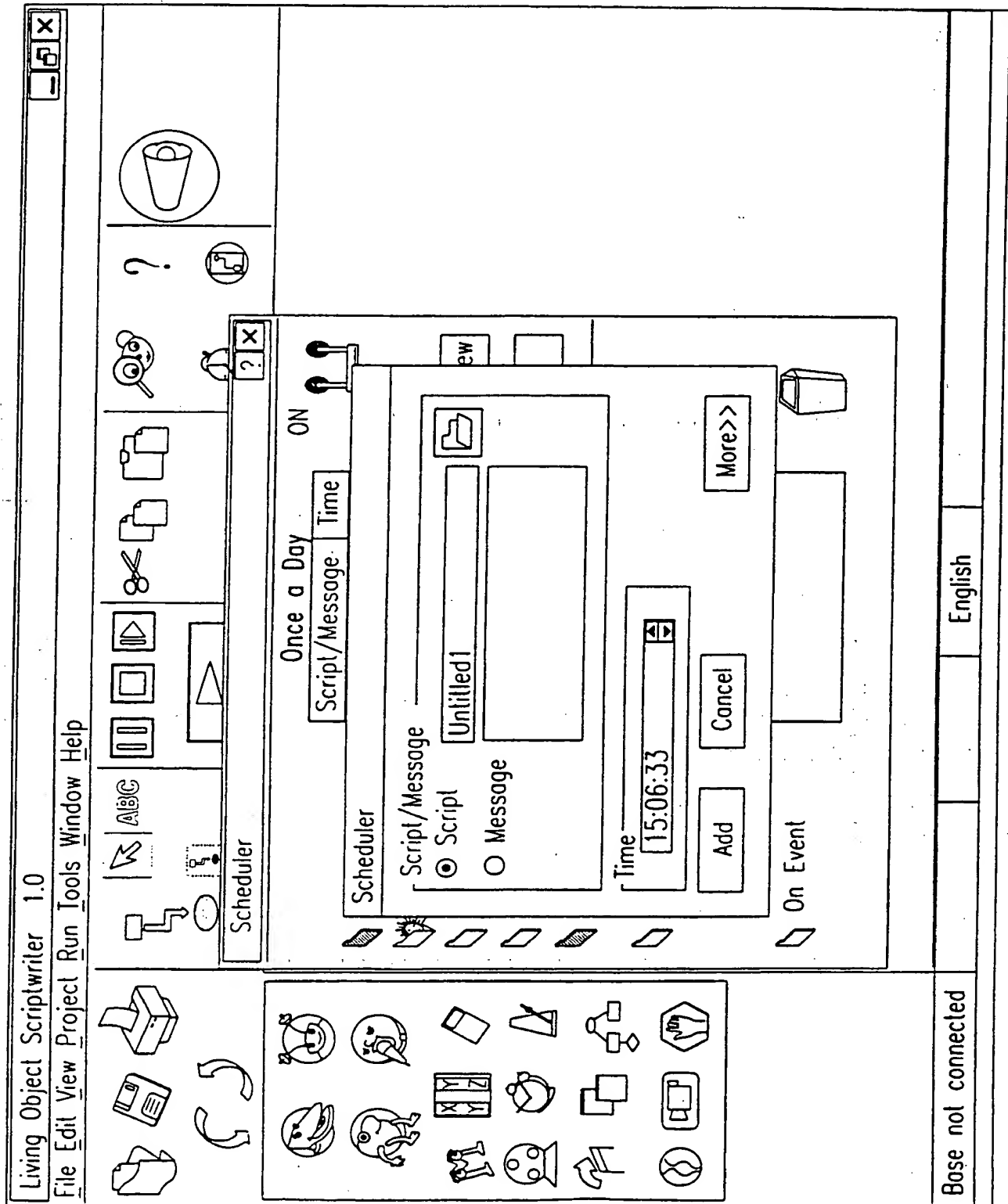


FIG. 117

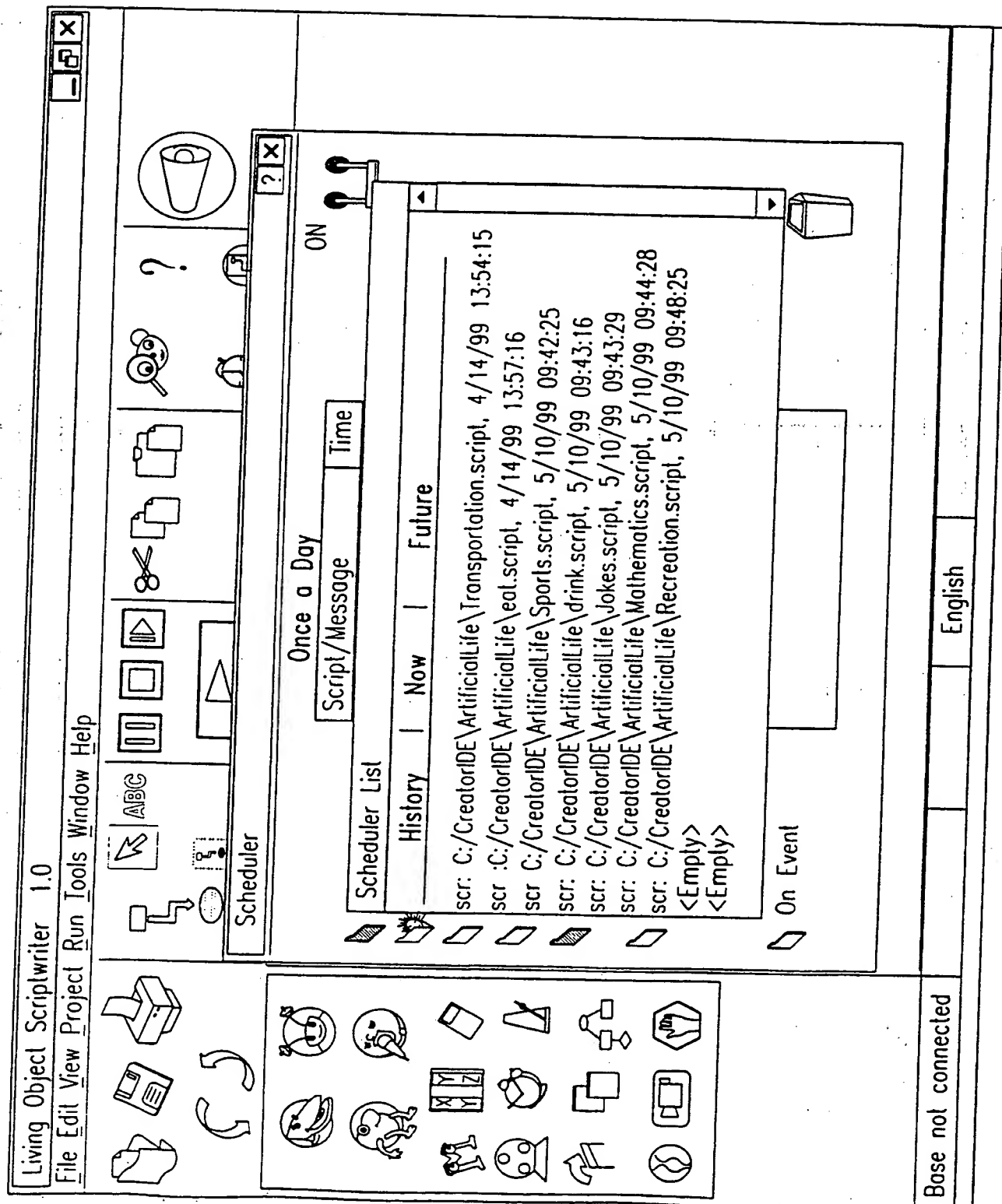


FIG. 118

113/120

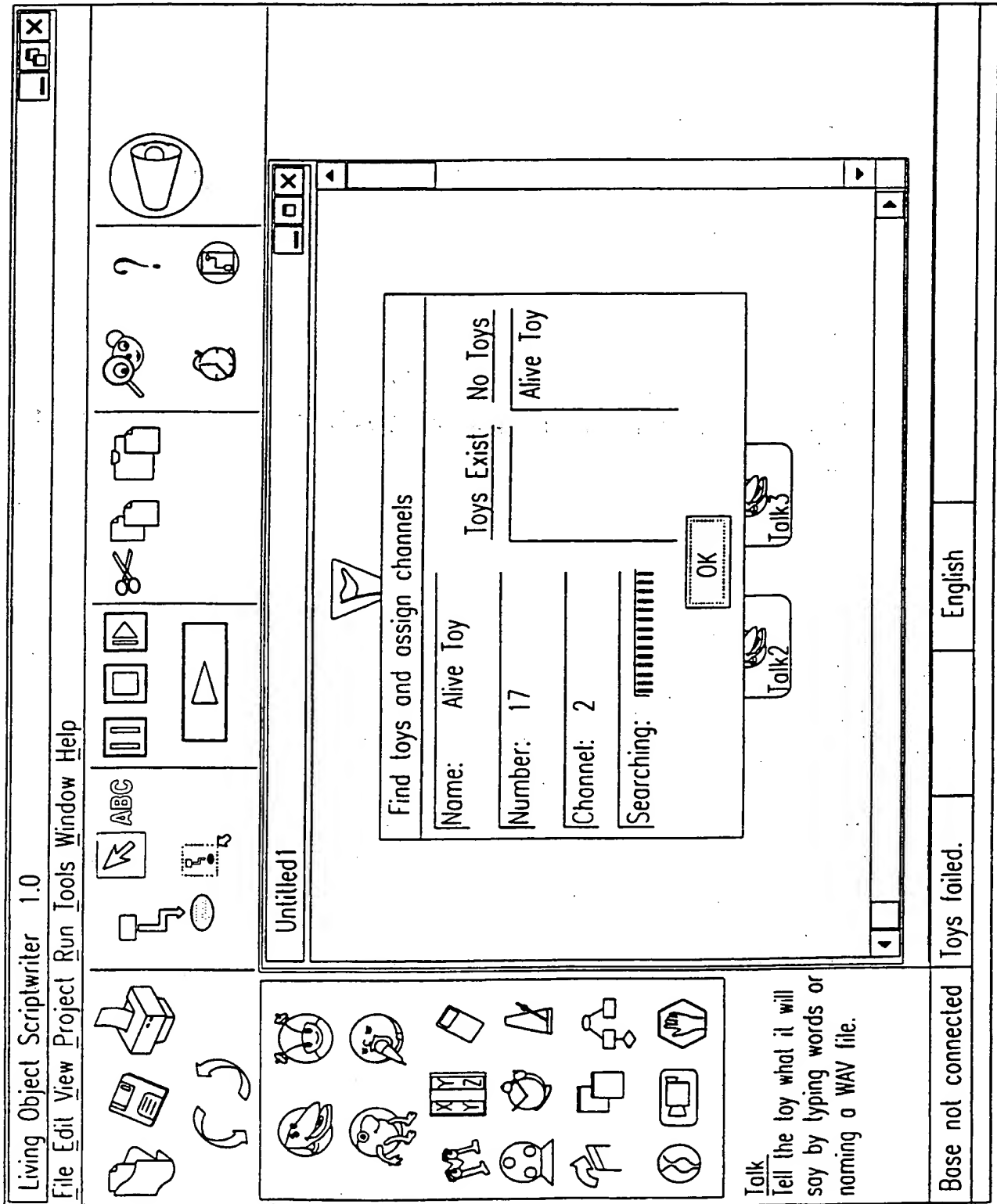


FIG. 119

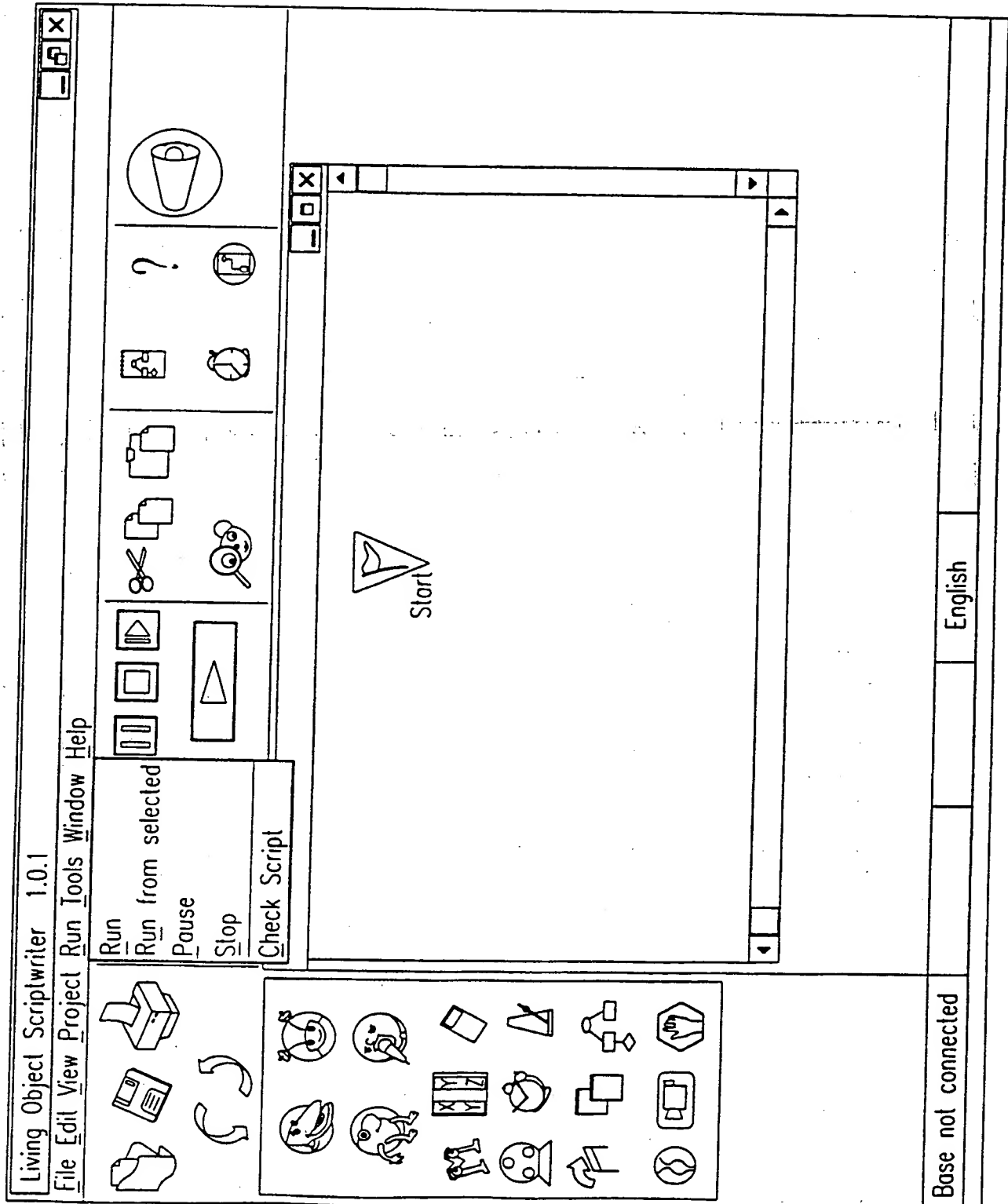
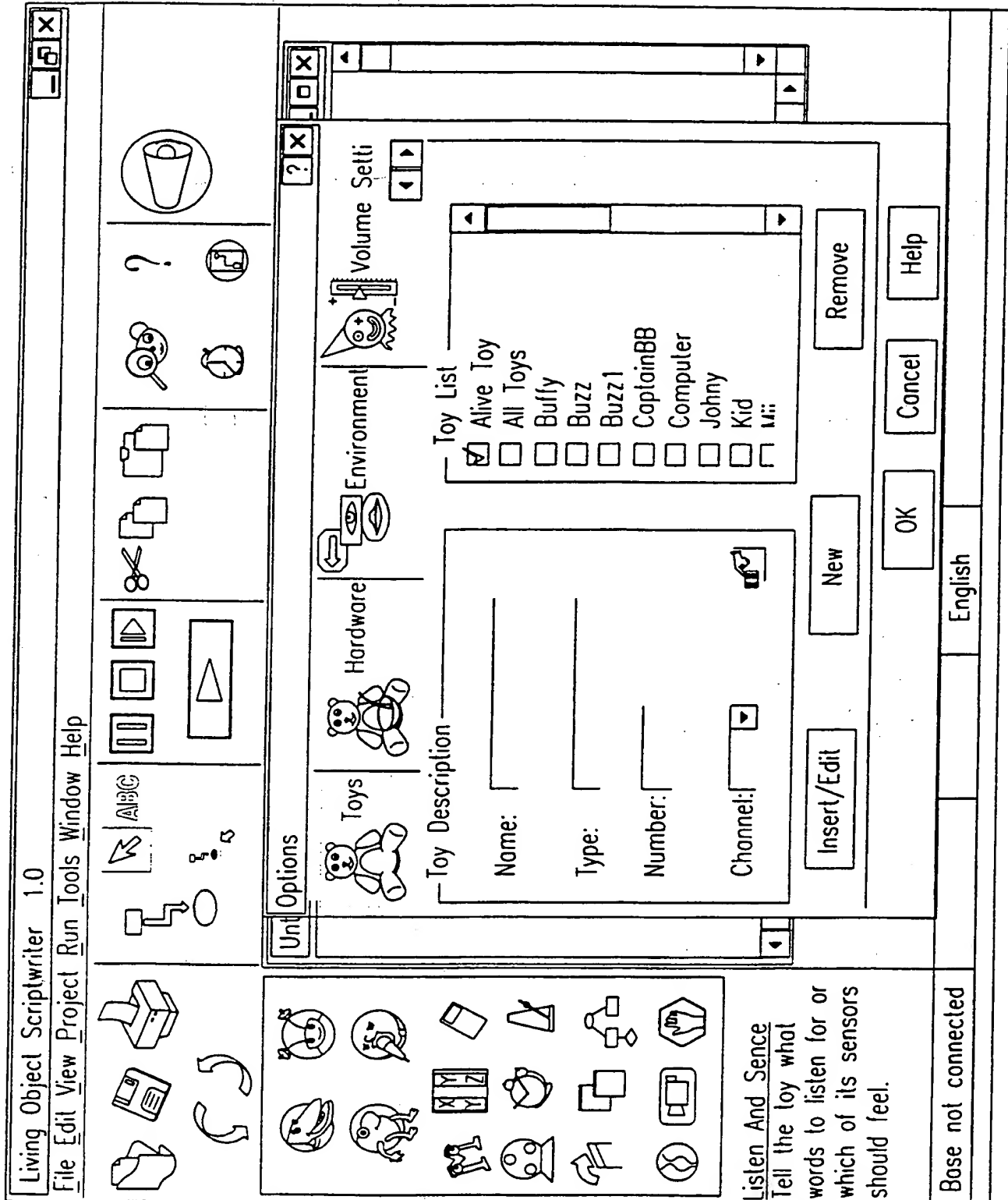


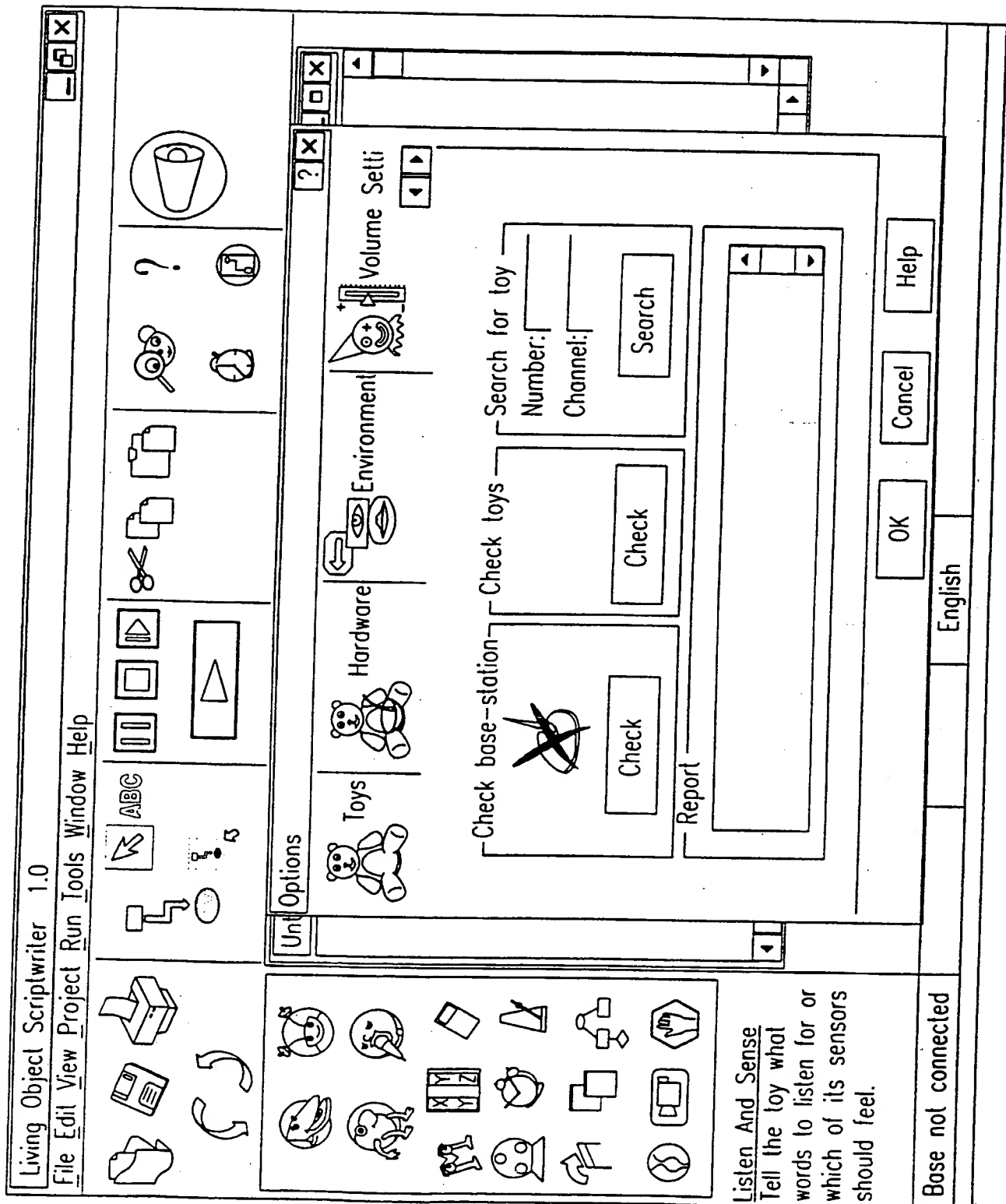
FIG. 120

115/120



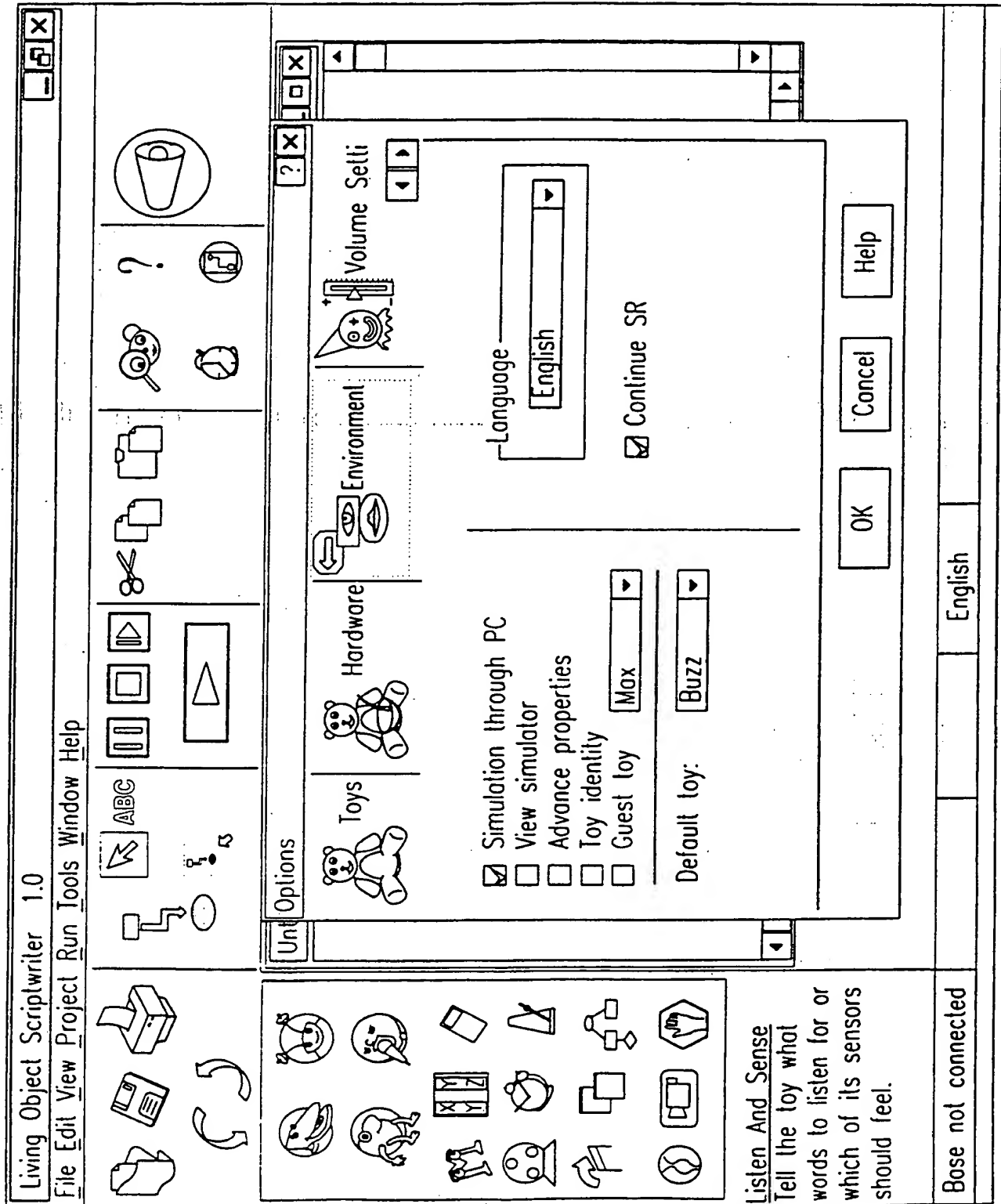
116/120

FIG. 121



117/120

FIG. 122



118/120

FIG. 123

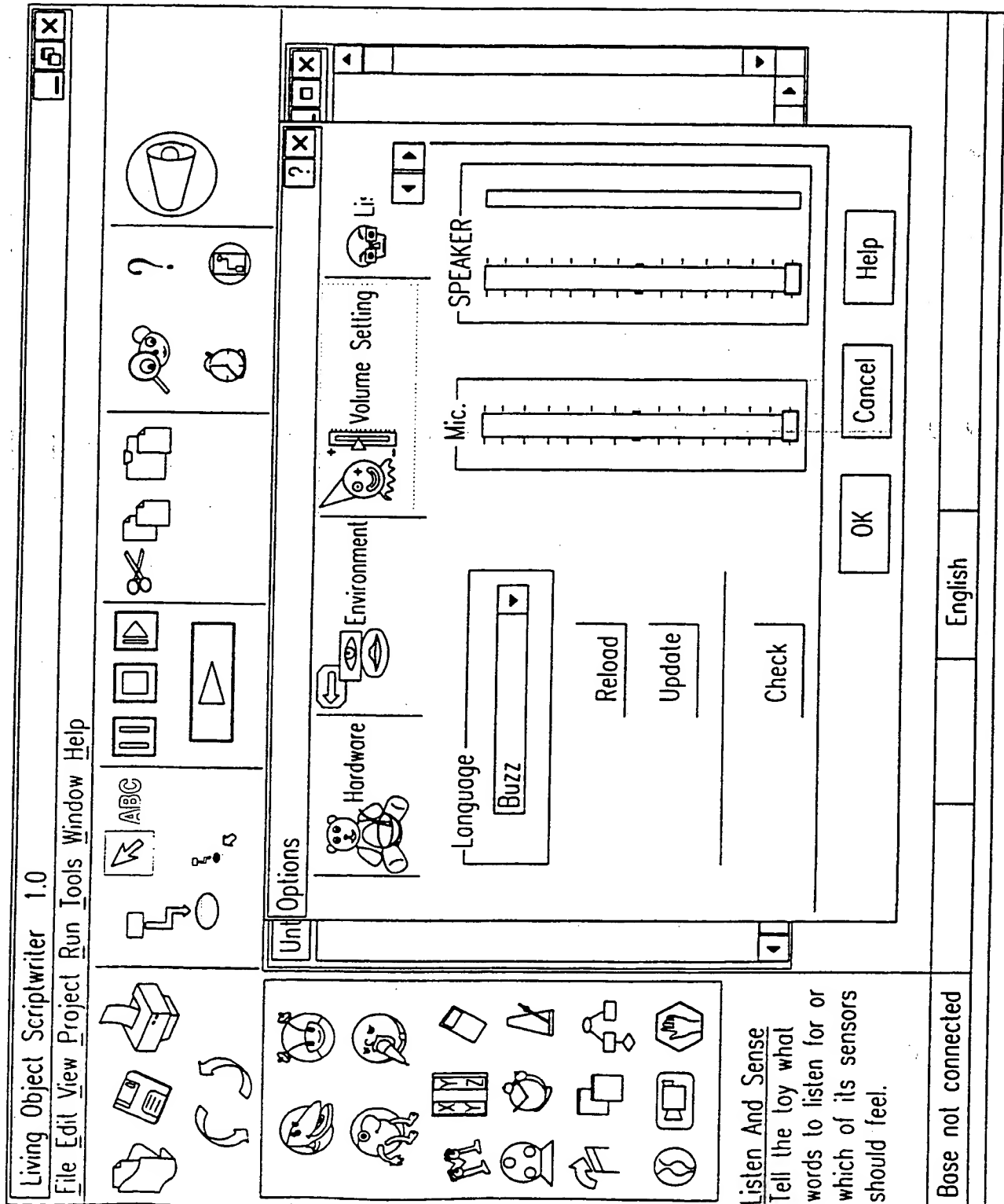


FIG. 124

119/120

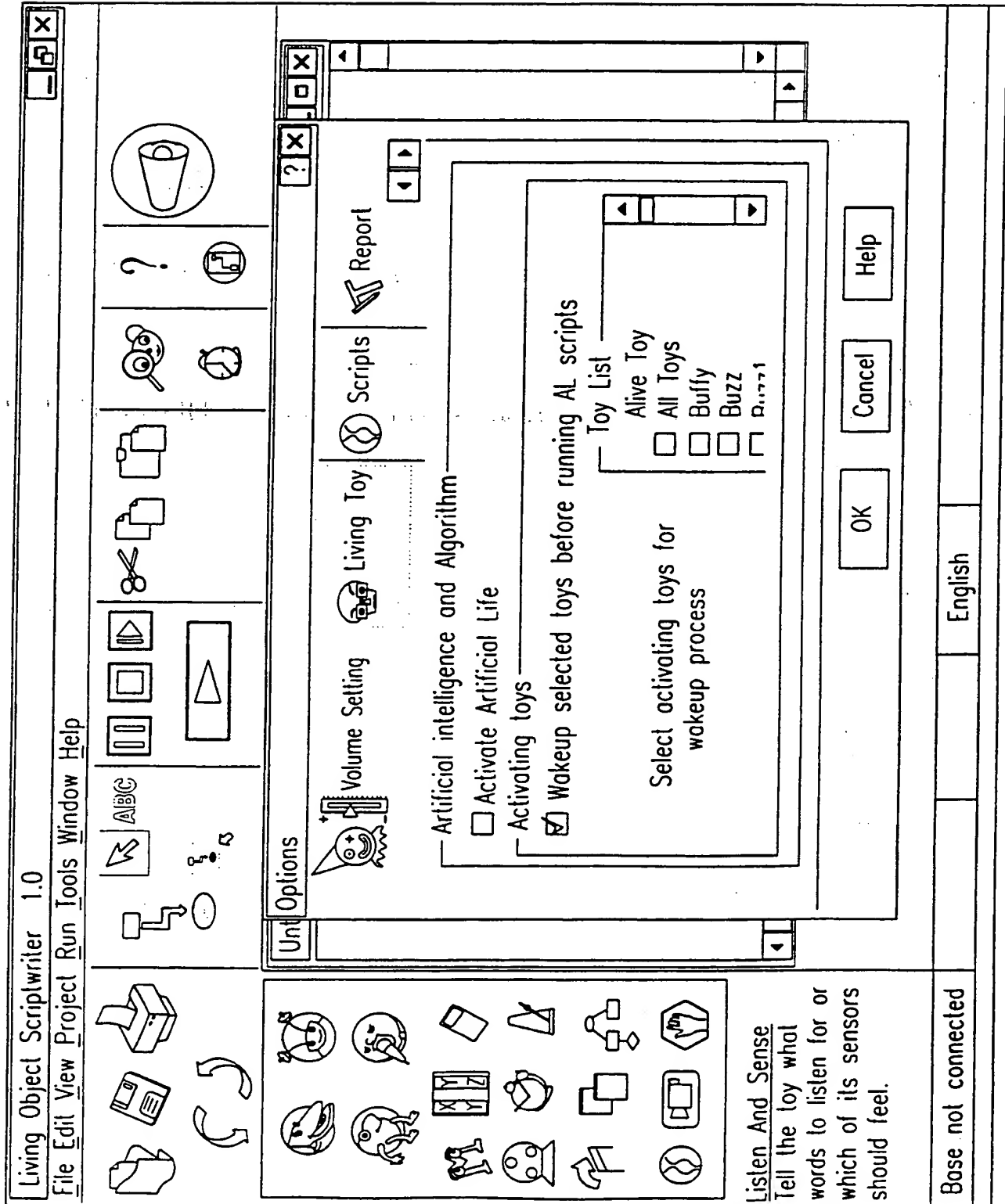
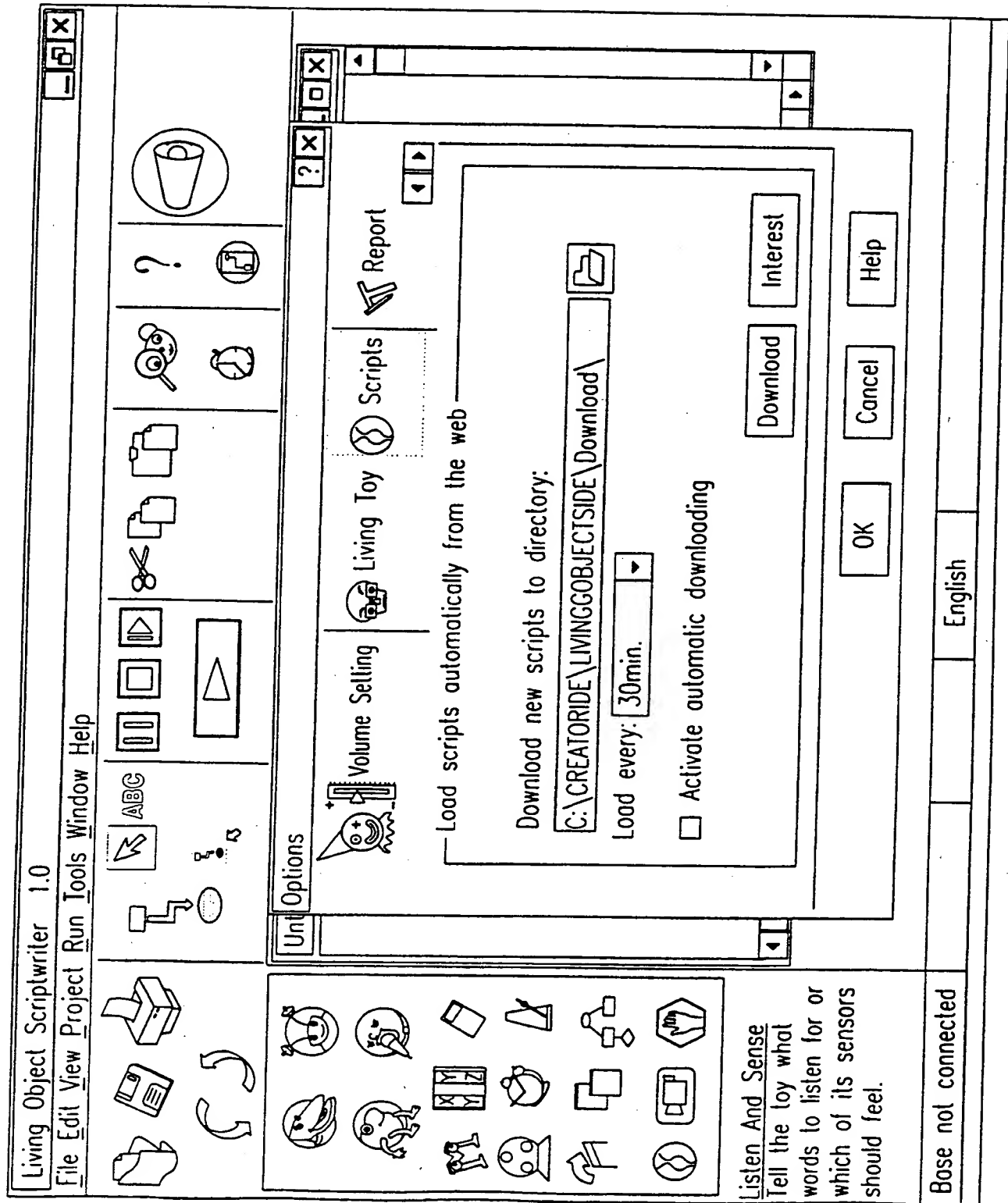


FIG. 125

120/120



INTERNATIONAL SEARCH REPORT

In .ational application No.
PCT/IL99/00637

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 3/00, 3/14

US CL :345/348, 349, 335, 357, 302, 967, 978

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/348, 349, 335, 357, 302, 967, 978

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

animate figure, pictorial image; animate script, interactive toys

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|-----------|--|-----------------------|
| Y,P | US 6,002,401 A (BAKER) 14 December 1999, col 17, lines 4-67. | 1-34 |
| Y | US 5,774,357 A (HOFFBERG et al) 30 June 1998, col 32, lines 7-56. | 1-34 |
| Y | US 4,884,972 A (GASPER) 05 December 1989, col 8, lines 5-50. | 1-34 |

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

| | |
|---|--|
| * Special categories of cited documents: | *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention. |
| *A* document defining the general state of the art which is not considered to be of particular relevance | *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone |
| *E* earlier document published on or after the international filing date | *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | *A* document member of the same patent family |
| *O* document referring to an oral disclosure, use, exhibition or other means | |
| *P* document published prior to the international filing date but later than the priority date claimed | |

Date of the actual completion of the international search

09 FEBRUARY 2000

Date of mailing of the international search report

23 FEB 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-6606

Authorized officer

RAYMOND J. BAYERL

Telephone No. (703) 305-9789

THIS PAGE BLANK (USPTO)